

凌阳科技

UNSP IDE 使用说明书

V1.0　2005-7-12

凌阳单片机推广中心

http://www.sunplusmcu.com

版权声明

目 录



第1章 入门

功能综述

μ'nSP® IDE 是由凌阳科技提供的一个集成开发环境，它集程序的编辑、编译、链接、调试和仿真等功能为一体。具有友好的交互界面、下拉菜单、快捷键和快速访问命令列表等，使程序设计工作更加方便、高效。此外，它的软件仿真功能可以不连接仿真板，模拟硬件的部分功能来调试程序。

手册从不同的方面向用户展示了μ'nSP® IDE 。如果用户需要更详细的信息，请和凌阳单片机推广中心保持联系。

安装程序

使用 μ'nSP® IDE 之前，用户必须运行 Windows98®/Windows2000®/WindowsXP®。注意：在 Windows2000®/WindowsXP®上使用本工具，用户必须拥有管理员权限。打印端口必须被设置为[SPP]（Standard Parallel Port）模式，端口地址建议被设置为 378H。

运行 setup.exe 程序后，安装程序被解压缩。显示画面如图：

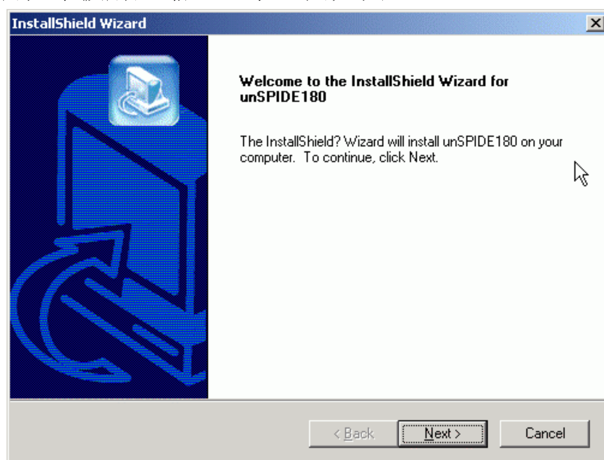


图 1

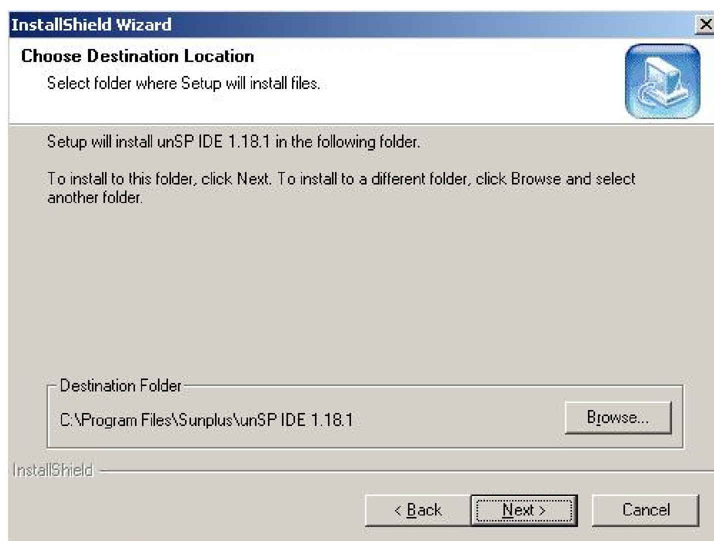


图 2

按照提示，安装程序就可以把μ'nSP® IDE 安装到硬盘上。用户可从 setup.exe 里定制安装内容。
典型(Typical)：本选项用于安装所有常用部件(默认项)。
精简(Compact)：本选项用于只安装所有基本部件。
自定义(Customize)：本选项用于安装用户自选部件。

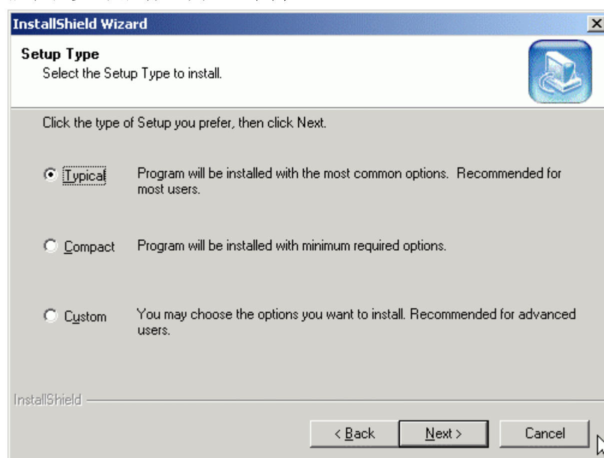


图 3

选择[开始]→[程序]→[sunplus]→[u'nSP IDE 1.18.1]，启动μ'nSP® IDE。



第2章 速成指南

从[开始]菜单内启动工具；

选择[File]→[Open Project]，在‘打开’对话框内选择所要打开的工程；

Workspace 窗口显示在工具的左半边，在这个窗口内，用户可以看到当前工程所包含的所有文件；

选择[Build]→[Rebuild All]，进行源文件的编译和链接。 编译链接过程里的错误显示在 Output 窗口内；

选择[Build]→[Start Debug]→[Download]，把程序加载到内存，然后，用户可以用 Debug 菜单内所提供的调试命令来调试和运行程序。选择[Build]→[Start Debug]→[Go]，在调试器内运行程序。

第3章 μ'nSP® IDE 窗口

工具界面

主界面包括三个主要窗口：工作区窗口(Workspace window)、编辑窗口(Edit window)和输出窗口(Output window)。只需在各窗口内单击鼠标左键即可把该窗口激活。此外，在主界面上，还提供工具栏等一些方便用户操作的工具。

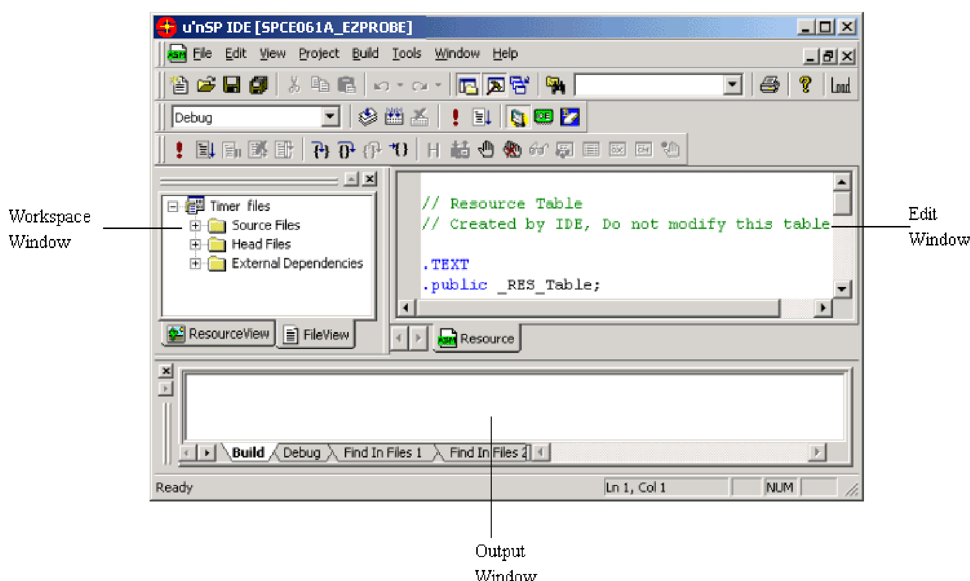


图 4 主窗口

Workspace 窗口

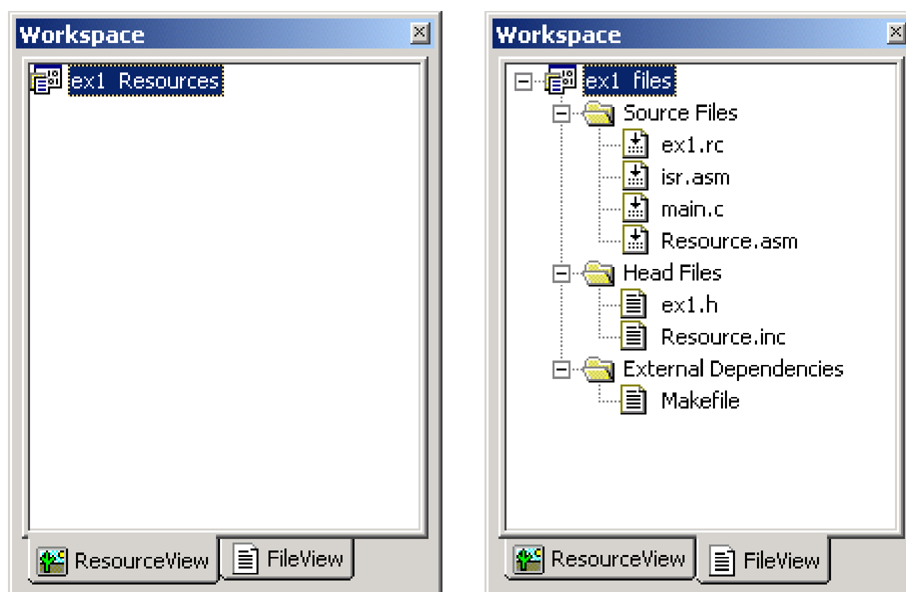


图 5 Workspace 窗口

在 Workspace 窗口内，用户可查看到当前工程所包括的全部文件。Workspace 窗口由 FileView 和 ResourceView 两个视窗组成。

单击 FileView 标签，用户可以方便浏览到工程内的各文件。FileView 视窗用层次图排列出当前工程的所有文件的逻辑关系。Files 文件夹包含了源程序、程序接口和说明硬件配置情况的文件。Resource 文件夹包括了各种资源文件(rc)。Source Files 文件夹用于保存源文件。Head Files 文件夹用于保存头文件。External Dependencies 文件夹用于保存对工程的一些标注信息。

ResourceView 视窗列出当前工程用到的所有资源。

可以单击视窗内分支顶部旁边的＋和－号展开和收缩层次图。

注意，Workspace 窗口所体现的逻辑关系不是指文件在硬盘上的物理位置，而是一个逻辑从属关系。用户可用拖曳的办法改变文件的逻辑位置。

在 Workspace 窗口内，不同类型的文件有不同的图标表现。

图标	作用
	源文件，经编译，生成目标文件。
	包含文件

Output 窗口

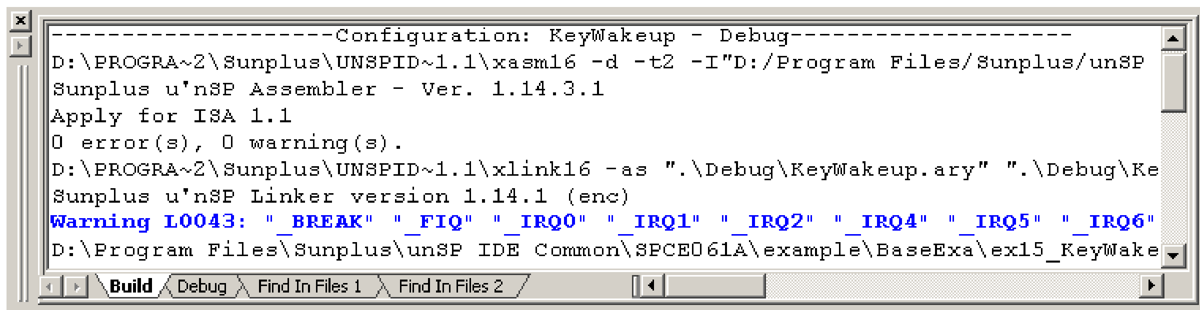


图 6 Output 窗口

Output 窗口用于显示编译、调试和查找的结果。

在窗口底部有几个视窗标签：Build、Debug 和 Find in Files 等。用鼠标单击这些标签，可以激活相应的视窗。

Build：显示编译和链接过程里产生的信息，包括文件编辑过程里的错误和警告信息等。
Debug：显示程序调试过程里出现的信息。
Find in Files：显示在文件中查找字符的结果。

Edit 窗口

在 Edit 窗口里，文件的打开格式有两种：用户可用文本格式打开文件，也可以用二进制代码格式打开文件。

文本编辑器

文本编辑器可以用来打开µ'nSP®汇编语言程序和 C 语言程序。

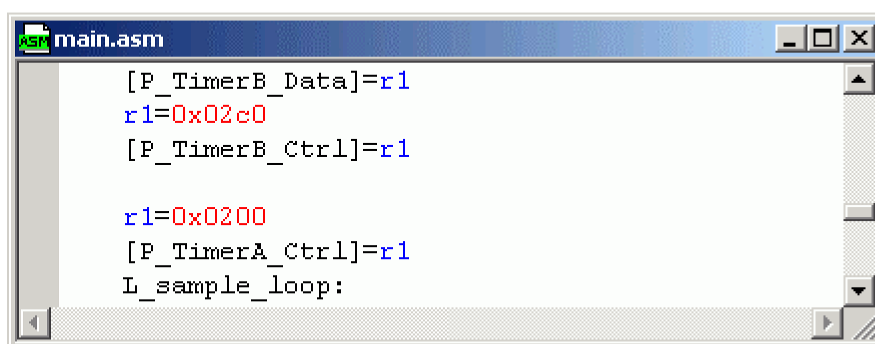


图 7 文本编辑器

二进制代码编辑器

二进制代码编辑器让用户在 Edit 窗口里以十六进制数/ASCII 字符的形式来编辑二进制代码的资源文件。

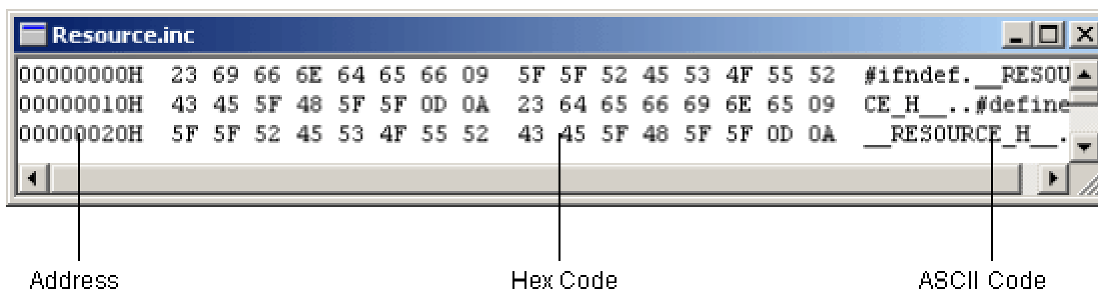


图 8 二进制代码编辑器

Debug 窗口

Memory 窗口

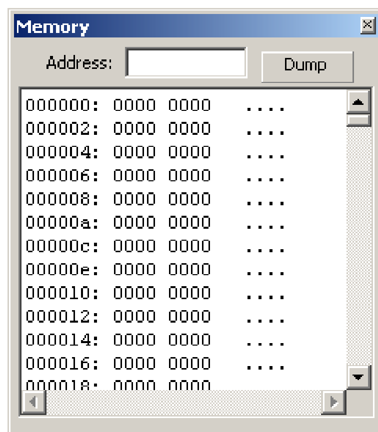


图 9 Memory 窗口 1

在调试过程里，可选择[View]→[Debug Windows]→[Memory]，打开 Memory 窗口。单击调试工具栏的 Memory 按钮，通过热键 Alt+2 也可打开、隐藏 Memory 窗口。

在 Memory 窗口内，存储单元的值以十六进制的格式显现。选择某数据后可直接改动内存单元的内容。在该窗口上方的 Address 文本输入框内输入一个内存单元地址，可很快查看到其中的内容。利用窗口内的滚动条，可以查看任一内存单元的值。

在 Memory 窗口内双击鼠标左键，可激活 Go to Address 对话框。在文本框内，输入地址，即可在 Memory 窗口内显现以输入地址为首行的连续的内存单元的值。



图 10 Go to Address 对话框

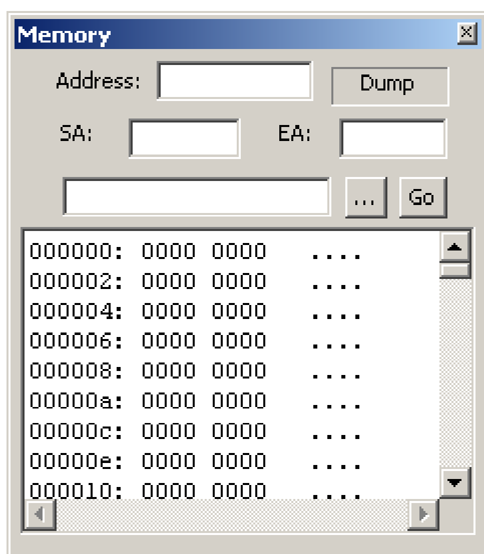


图 11 Memory 窗口 2

Memory 窗口提供了一个转存功能，在 Memory 窗口单击[Dump]，可激活如上图的窗口。利用 Memory 窗口的转存功能，可以将指定地址范围内的内存数据存储到文件内。SA 和 EA 可用于指定地址范围，文件选项用来选择文件。填写好上述内容后，单击[Go]，执行程序转存。

Register 窗口

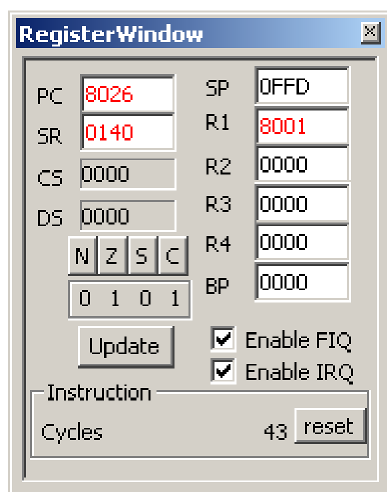


图 12 Register 窗口

在调试过程里，可选择[View]→[Debug Windows]→[RegisterWindow]打开 Register 窗口。单击调试工具栏的 Register 按钮，通过热键 Alt+3 也可打开、隐藏 Register 窗口。

Register 窗口显示通用寄存器和特殊寄存器的内容。在 Register 窗口，用户可以根据需要改动各寄存器的值、中断标志位的值，控制中断响应，还可以查看指令执行的周期数。改动数据后按[Update]来更新。选择[Reset]，让 CPU 复位。

注意，当连接仿真板后，在 Register 窗口看不到指令周期数。



Command 窗口

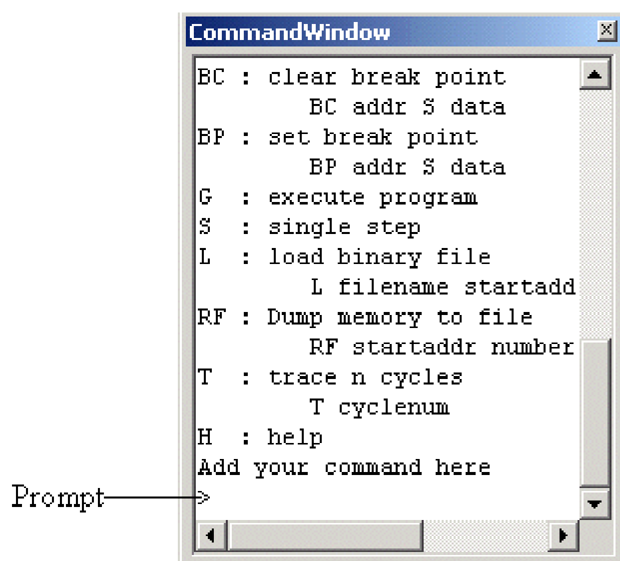


图 13 Command 窗口

在调试过程里，可选择[View]→[Debug Windows]→[CommandWindow]打开 Command 窗口。单击调试工具栏的 Command 按钮，通过热键 Alt+4 也可打开、隐藏 Command 窗口。

在文本框内输入 H 命令后按回车，所有命令和相应的使用方法就列在窗口内，用户可拖动窗口右侧的滚动条，查看所需的命令。命令在输入后立即被执行。

Command：

Q ：	退出μ'nSP® IDE
Dump：	以字为单元转存存储区的数据 Dump <起始地址> <转储字数> dump 100 100 //转存 0x100 ~ 0x1ff 范围内的数据
EF：	允许产生 FIQ 中断(缺省)
DF：	禁止产生 FIQ 中断
EI：	允许产生 IRQ 中断(缺省)
DI：	禁止产生 IRQ 中断
SN：	设置负标志
NN：	清除负标志
SS：	设置符号标志
NS：	清除符号标志
SZ：	设置零标志
NZ：	清除零标志

SC： 设置进位标志

NC： 清除进位标志

X： 复位

RX： 设定寄存器的值

Rx <寄存器号> <设定值>.

Rx 3 abcd //R3 的值设为 0xabcd

O： 设定 RAM 单元的值

O <内存地址> <设定值>

O 7016 abcd //0x7016 单元的值设为 0xabcd

F： 设定一 RAM 块的值

F <内存起始地址> <设定数目> <设定值>

F 100 100 1234 // 0x100 ~ 0x1ff 单元填入 0x1234

BC： 清除断点

BC <断点地址> <断点标志> <断点数据>

BC 8000 8082 1234

//清除当向 0x28000 单元内写入数据 0x1234 时的条件断点

BP： 设置断点

BP <断点地址> <断点标志> <断点数据>

BP 8000 8082 1234

//设置当向 0x28000 单元内写入数据 0x1234 时的条件断点

G： 开始执行程序

S： 单步运行程序

L： 把二进制文件加载到内存

L <文件名> <起始地址> <结束地址> <内存的起始地址>

L test.bin 100 1ff 8000

//将 test.bin 文件第 0x100~0x1ff 单元的数据加载到 0x8000 内存单元

RF： 将内存数据转存到文件内

RF <起始地址> <内存单元个数> <文件名>

RF 100 100 test.bin

//将 0x100 ~ 0x1ff 单元的内容转存到 test.bin 文件内

T: 跟踪 n 个周期

T 周期数

H: 显示µ'n SP® IDE 的所有命令和相应的内容描述

Breakpoint 窗口

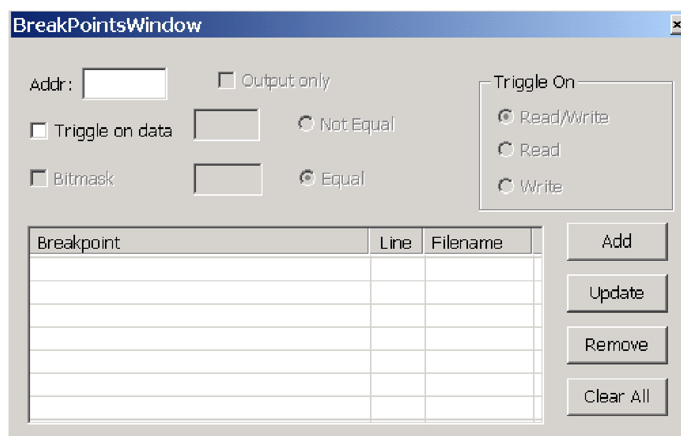


图 14 Breakpoint 窗口

在调试过程里，可选择[View]→[Debug Windows]→[BreakPointsWindow]打开 BreakPoint 窗口。单击调试工具栏的 BreakPoint 按钮，通过热键 Alt+5 也可打开、隐藏 BreakPoint 窗口。

在 BreakPoint 窗口内，用户可任意改动程序内的断点。断点信息是工程的不可缺少的内容。

Addr：欲被设置断点的地址；

Output Only：在连接仿真板运行时，当程序执行到断点位置后，向指定管脚输出一个脉冲信号；

Triggle on Data：这是一个数据过滤器。选择 Triggle on Data 和 Equal(Not Equal)，当程序执行到断点位置后，自动检查断点地址的数据和 Triggle on Data 文本框内所指定的数据是否相等。如果相等(不等)的条件满足，程序在断点地址被中断；

Bitmask：用于屏蔽断点地址内的某些位；

Triggle on Write&Read：当对数据进行存取时，触发中断。

Watch 窗口

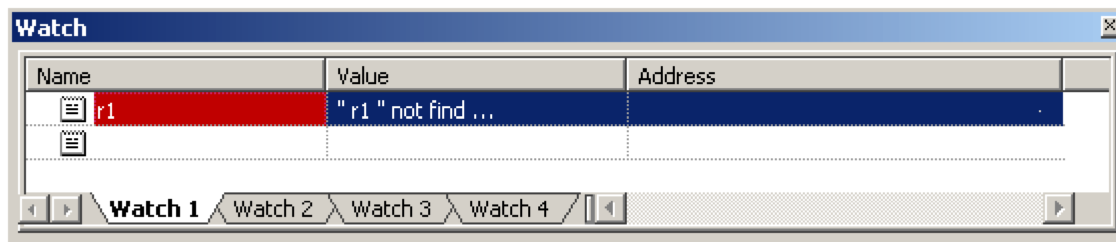


图 15 Watch 窗口

在调试过程里，可选择[View]→[Debug Windows]→[Watch]打开 Watch 窗口。单击调试工具栏的 Watch 按钮，通过热键 Alt+C 也可打开、隐藏 Watch 窗口。

Watch 窗口用于查看和更改变量值。

Watch 窗口由 **Watch1, Watch2, Watch3,** 和 **Watch4** 视窗组成。每一个视窗利用数据表来显示变量的值。

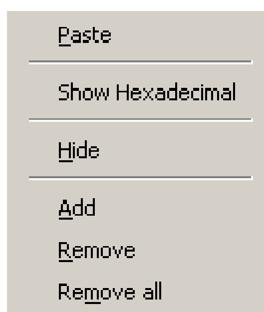


图 16 Watch 热键

在 Watch 窗口单击鼠标右键，可以激活相应的热键菜单。利用 **Paste, Show decimal/hexadecimal, Hide, Add, Remove** 和 **Remove all**，用户可改变数据的显示格式、添加和删除数据、控制 Watch 窗口的显隐。

Disassembly 窗口

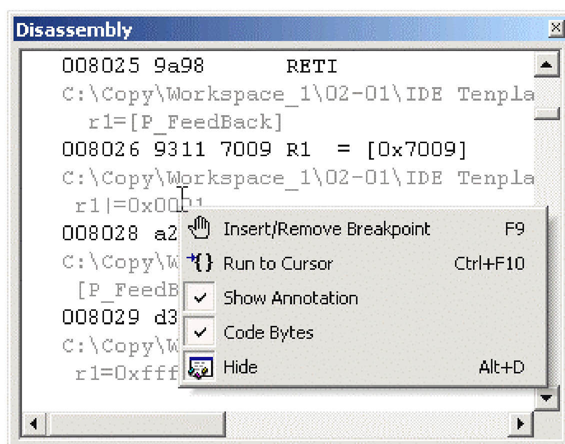


图 17 Disassembly 窗口

在调试过程里，可选择[View]→[Debug Windows]→[Disassembly]打开 Disassembly 窗口。单击调试工具栏的 Disassembly 按钮，通过热键 Alt+D 也可打开、隐藏 Disassembly 窗口。

Disassembly 窗口的内容随 PC 指针的值而变化。在 Disassembly 窗口上双击鼠标左键，可以激活 Goto Address 对话框。

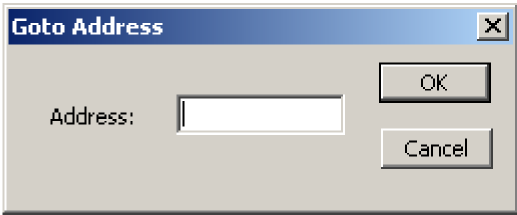


图 18 Goto Address 对话框

在 Address 文本框内输入地址，程序从指定的地址行开始显现反汇编指令。

在 Disassembly 窗口单击鼠标右键，可以激活相应的热键菜单。热键菜单可以控制程序的执行、代码的显示格式和 Disassembly 窗口的显隐。

两种查看程序源代码的办法：
选择[Tools]→[Option]→[Debug]→[Source annotation]；
在热键菜单内选择[Show Annotation]。

利用相同的办法，用户选择[Code Bytes]，可以查看到操作码。

History Buffer 窗口

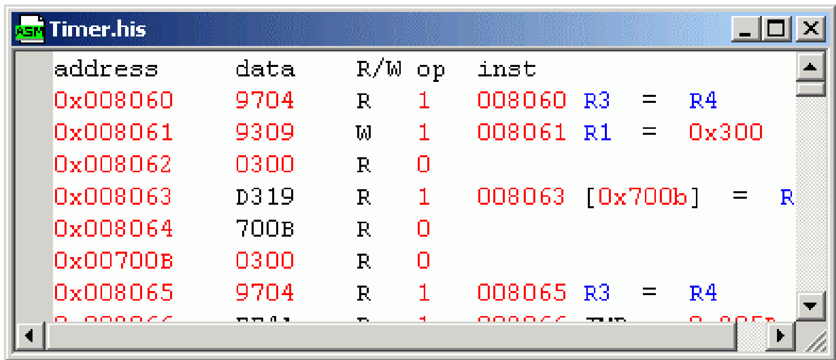


图 19 History Buffer 窗口

History Buffer 窗口显示程序运行的各个记录。

选择[Project]→[Setting]，在对话框的 General 页内选择 Simulator 和 PC Trace Enable。在执行完程序后，被执行的指令、状态等将被存储到历史缓冲区，以*.his 保存。

在 General 页内选择 Save Instruction Only 后，只用于记录被执行的指令。

文件执行后，单击H按钮，即可打开 History Buffer 窗口(历史缓冲区窗口)，被调试的程序的运行状况显示在 History Buffer 窗口内。



第4章 工程

工作区窗口

工作区窗口为开发人员提供了当前工程内的文件的各种信息：
程序源文件、名称和文件之间的逻辑关系；程序所需库文件清单；当前工程的各种设置。

工程的操作

工程内各类文件

工程文件（spj)：以'spj'为扩展名的工程文件包括创建一个工程所需的各种信息，以前版 IDE 的工程文件扩展名为'scs'；
资源文件（rc)：扩展名为 'rc'的资源文件包括当前工程的所有资源的信息；
资源表文件(asm)和资源表头文件(inc)；
Make File 文件；
C 语言包含文件(h)。

创建工程

µ'nSP® IDE 运行之后，可以开始创建程序的工程。工程包括创建一个特定程序所要的各种信息。

选择[File]→[New]， 打开 New 对话框，选择 Project 标签；

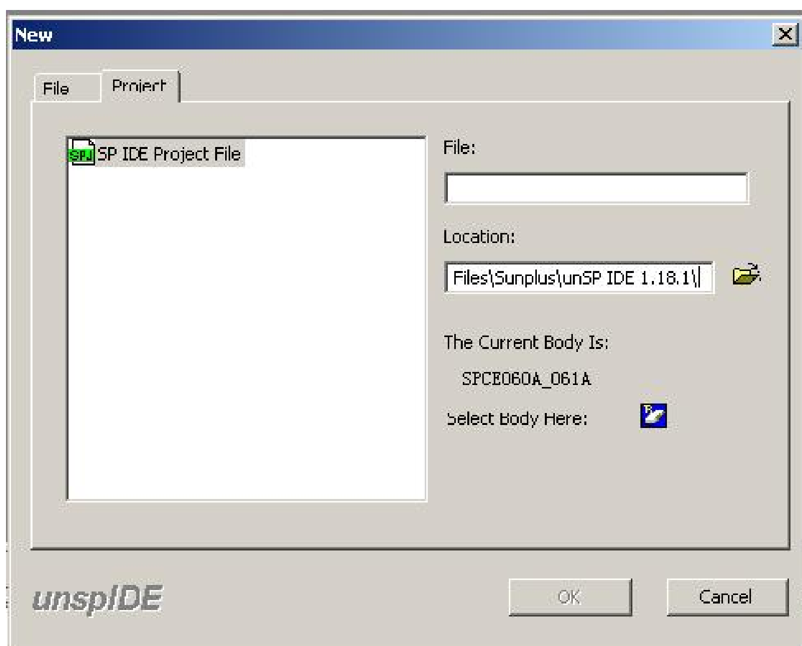


图 20 New 对话框

在 File 文本框内输入工程名称；
在 Location 文本框内输入工程文件的路径；
在 Select Body Here 区域内选择 Probe；
单击[OK]，创建工程。

这时，µ'nSP® IDE 生成一个新工程，工程信息在 Workspace 窗口内显示。如果 Workspace 窗口没有显现，可以选择[View]→[Workspace]，打开 Workspace 窗口。

生成程序源文件

选择[File]→[New]，打开 New 对话框：

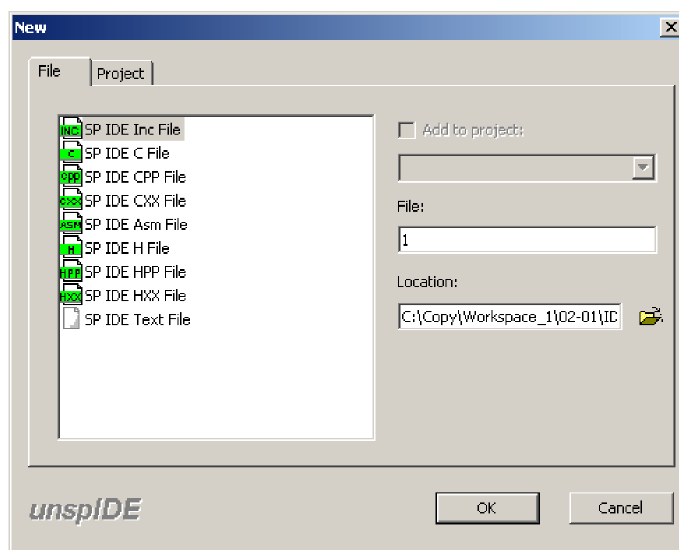


图 21 New 对话框

2. 在 File 页的文件类型清单里选择 SP IDE C File；
3. 在 File 框内输入源文件名称，Location 框内包含了生成文件时指定的工程文件夹路径；
4. 选择 [OK]。

选择 Add to project，可把源文件添加到某个工程内。

输入程序代码后，选择[File]→[Save]，可保存输入的内容。

向工程添加文件和资源

可通过两种办法向工程添加文件：

选择[Project]→[Add to Project]→[Files/Resource]，激活 Add Files 对话框；
在工程窗口的文件夹上单击鼠标右键，在热键菜单里选择 Add File To Folder，激活 Add Files 对话框。



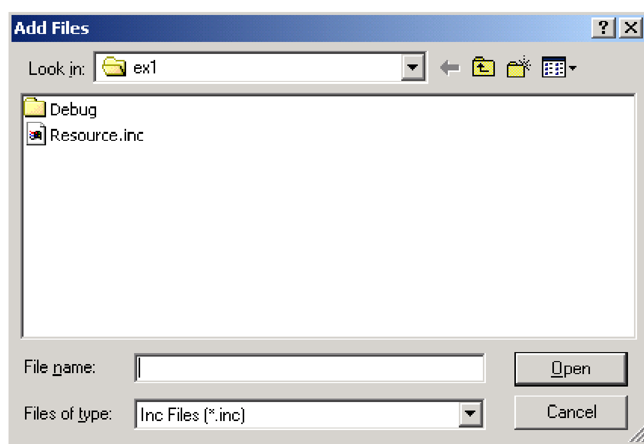


图 22 Add Files 对话框

在 Add Files 对话框的文件列表内选择文件，然后选择[Open]，向工程添加文件。

打开工程

一个工程可以按照前一次的设置再次被打开。µ'nSP® IDE 可以利用 Open Project、Open 和 Recent Project 菜单打开工程。

Open Project：
选择[File]→[Open Project]，打开 Open 对话框；
在 Type 内选择 Project Files(*spj)/(*scs)/Pack File(*pak)；
在文件列表内选择工程，单击[Open]。

Recent Project：
利用[File]→[Recent Project]，打开最近访问的工程。

Open：
选择 [File]→ [Open]，打开 Open 对话框；
在 Type 内选择 Project Files(*spj)；
在文件列表内选择工程，单击[Open]。

Open 对话框的 Preview 选项可用来预览文本文件。

保存工程

选择 [File]→[Save Project]，保存工程；
重编制某工程后，系统可保存工程。

创建目标文件

目标文件指工程的二进制可执行输出文件。µ'nSP® IDE 依据编译器和链接器的设置来生成工程的最终目标文件（S37、TSK)。

用户可任意在 Build Toolbar 工具栏内选择 Debug 和 Release，来创建调试版和发布版的输出文件。

使用工程内的元组

创建工程后，在 Workspace 窗口的 FileView 视窗内可以看到三个元组：Source Files、Head Files 和 External Dependencies。这是由系统自动创建的，不允许删除和移动。除了这三个元组外，用户可按需要通过热键等途径对元组进行添加、删除、移动、复制和重命名等操作。

删除和移动元组，意味着这个操作从逻辑关系上删除和移动元组内的所有的文件。

向工程添加元组

在工作区窗口内，选择某文件夹，然后按鼠标右键激活热键菜单；
选择 New Folder，激活 New Folder 对话框：

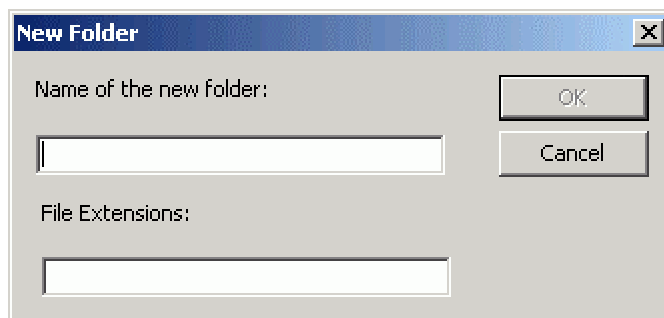


图 23 New Folder 对话框

在 Name of the new folder 文本框内输入元组名称，在 File Extensions 文本框内指定当前元组内文件的扩展名，选择[OK]。

设置完成后，当选择[Project]→[Add to Project]→[Files/Resource]向工程添加文件时，文件可自动保存到相应的元组内。

改变元组属性

工作区窗口内的各个元组都可以通过热键菜单改变属性。

在 FileView 视窗内的元组属性如图：



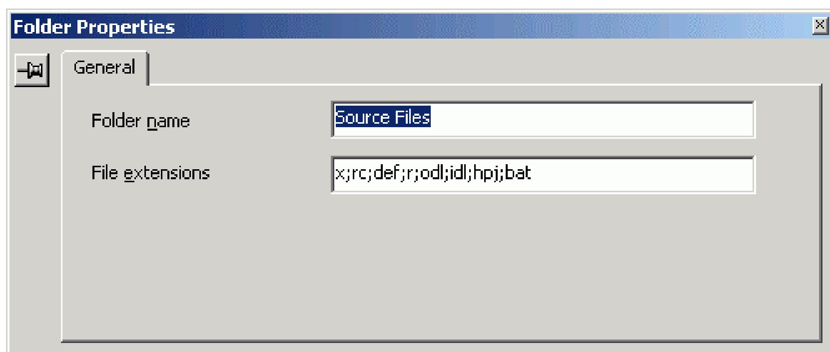


图 24 Folder Property 对话框 1

在 ResourceView 视窗内的元组属性如图：

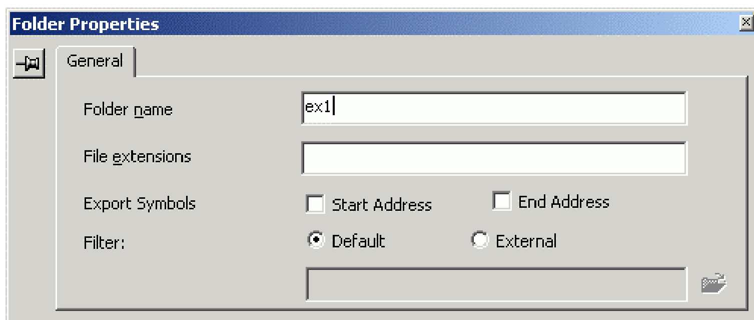


图 25 Folder Property 对话框 2

Folder Property 对话框用于改变当前元组的名称(Folder name)、可包含文件类型(File extensions)、输出符号表(Export Symbols)，选择相应的过滤器(Filter)。过滤器设置好后，可在链接之前将资源转换为 Sunplus 格式。

外部过滤器的格式：

Filter.exe <输入文件> <*输出文件>　　*输出文件的扩展名必须是 sp

选择 Probe 型号

选择 [Project]→[Select Body]，打开 Select Body 对话框，然后用户可以选择所需的 Probe 型号和 DLL。

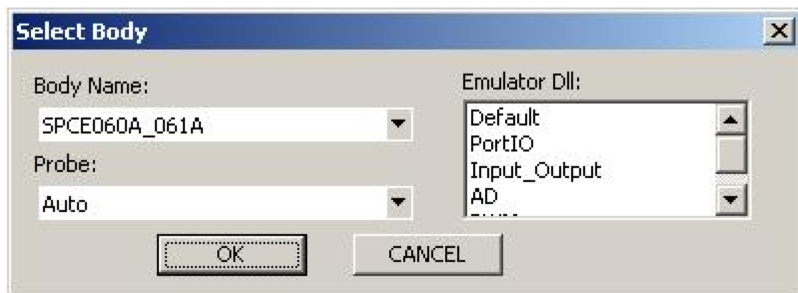


图 26 Select Body 对话框

Body Name 用于选择 Probe 型号，Version 用于选择仿真工具的版本，Emulator DLL 用于选择相应的周边动态链接库。这些信息和工程是息息相关的。

工程设置

在编制工程之前，可改变影响工程源文件编译、链接的设置。

为改变设置，选择[Project]→[Setting]，打开设置对话框。 μ'nSP® IDE 的多数设定只针对工程一级，可以针对不同目标对开发环境的各个要素进行设置。 通常，μ'nSP® IDE 可创建两个版本的工程：调试版和发布版。针对工程级的设置同时适用于工程内的所有文件。

指定输出文件的路径

在μ'nSP® IDE 内，用户可为各个目录指定编译过程中的中间文件和目标文件的存取目录。

1.选择 [Project]→[Setting]，打开 Setting 对话框：

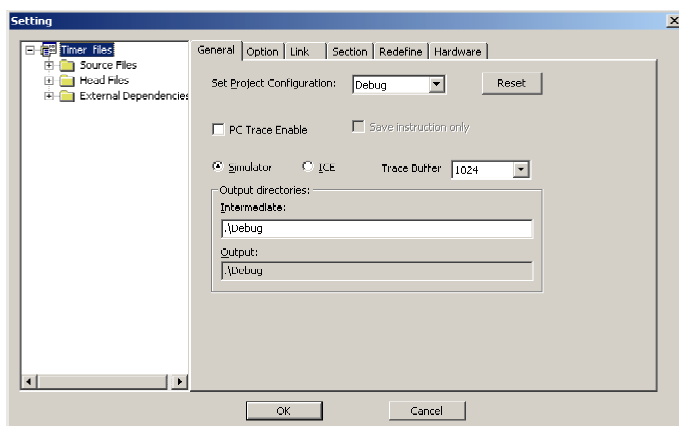


图 27 Setting 对话框

2.选择 [General]；
3.在 Intermediate 框内输入相对路径；
4.选择 [OK]。用户可看到用户设定的相对路径自动写入 Output 文本框；
5.在选择 [Build]→[Build]或[Build]→[Rebuild All]后，编译过程中的中间文件和目标文件将被存放到用户所指定的目录里。

工程设置

打开一个工程；
选择 [Project]→[Setting]，打开 Setting 对话框；
选择对话框上部的各个属性标签。

[General]：



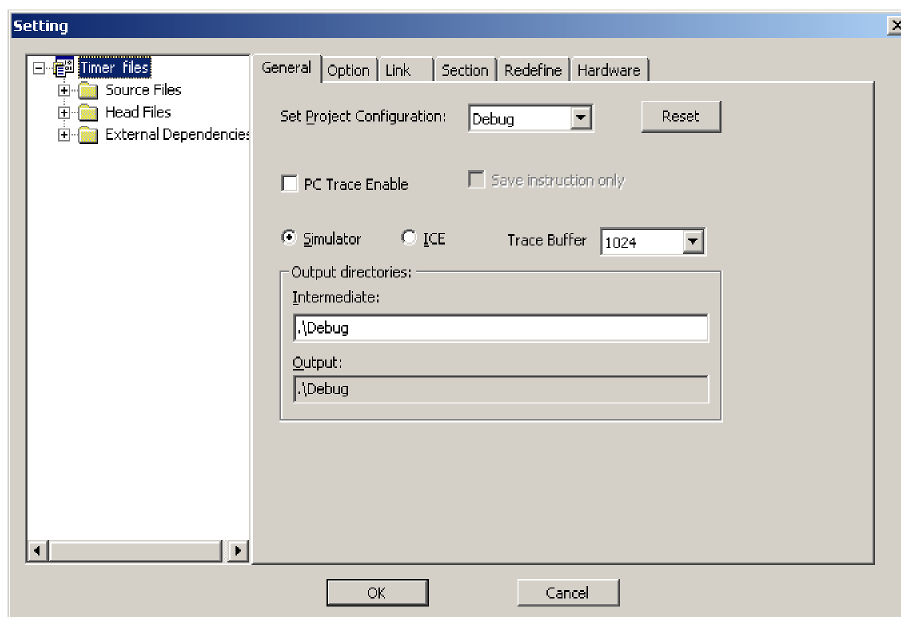


图 28 General 属性页

在 General 页里，可以完成µ'nSP® IDE 的一些基本设置。

Set Project Configuration： 选择所创建工程的设置，有两种选择：Debug 和 Release；

Simulator/ICE：选择µ'nSP® IDE 运行模式。选择 ICE，用户必须通过打印端口把仿真板和电脑相连接。选择 Simulator，可利用µ'nSP® IDE 提供的软件仿真功能，所有数据被存入缓存；

PC Trace Enable：激活 PC Trace 功能；

Save instruction only：激活此项，可保存运行过程里所有用到过的指令。未选中，µ'nSP® IDE 保存更多的运行相关信息(如：操作码等)至历史缓存区；

TraceBuffer Size：指定用于记录仿真过程里的缓存的容量；

Intermediate：指定编译过程里产生的临时文件的存储路径；

Output：指定目标文件的存储路径。通常，目标文件的存储路径和临时文件的存储路径相同，用户只需指定临时文件的存储路径即可；

Reset：复位所做的设置。

[Option]：

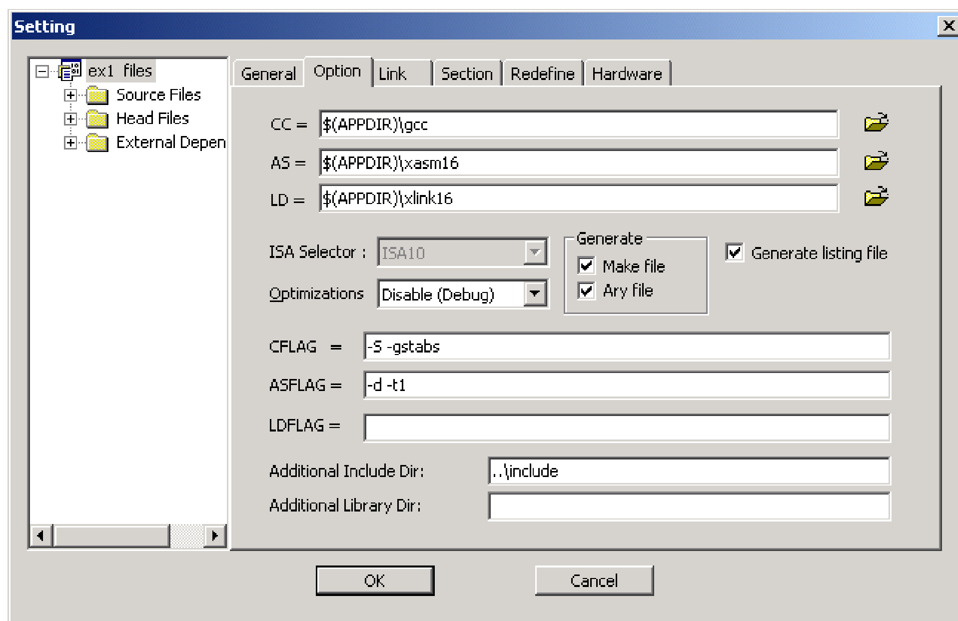


图 29 Option 属性页

在 Option 页里，可对µ'nSP® IDE 编译链接所需的软件工具进行设置。

CC：指定 C 编译器程序在 PC 机硬盘上的位置及其文件名；

AS：指定汇编器程序在 PC 机硬盘上的位置及其文件名；

LD：指定链接器程序在 PC 机硬盘上的位置及其文件名；

CFLAG： 指定 C 编译器运行及代码优化标志；

ASFLAG： 指定汇编器运行标志；

LDFLAG： 指定链接器运行标志；

Optimizations： 选择用户所需的代码优化类型 CFLAG 的优化标志随之改变;

ISA Selector：显示指令集版本；

MakeFile 选中该项以自动更新 makefile 文件；

Ary File 选中该项以自动更新.ary 文件；

Generate listing File 选中该项以自动更新.lst 文件；

Additional include dir：指定包含文件的路径；

Additional library dir：指定库文件的路径。



在 Debug 和 Release 模式里，CFLAG, ASFLAG, LDFLAG 可被指定不同的参数。系统根据用户选择的不同的模式，自动更改参数。

[Link]：

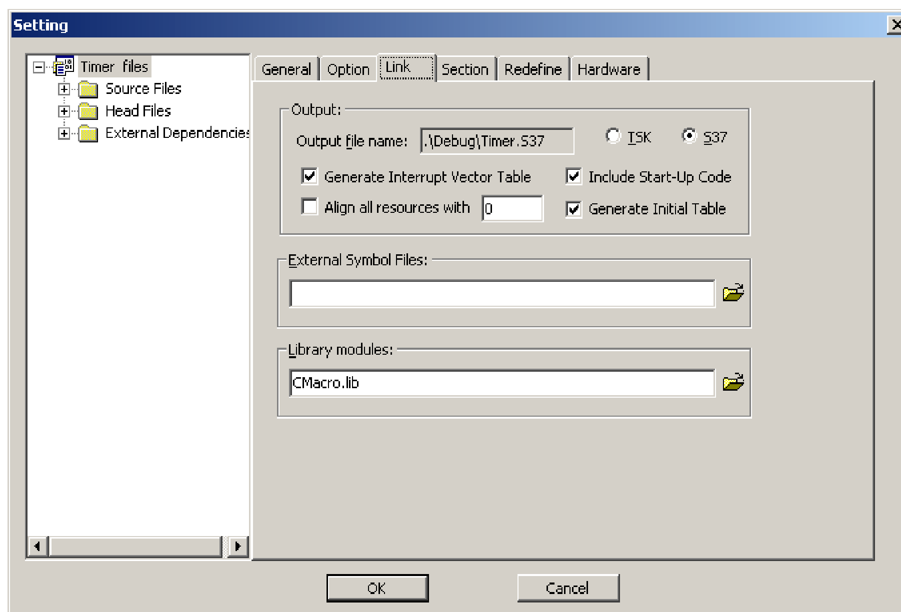


图 30 Link 属性页

在 Link 页里，可对链接器进行设置。

Output file name：指定二进制输出文件名：TSK/S37 为选择的二进制输出文件格式；

TSK/S37：两种目标文件类型。TSK 为二进制文件，S37 为 Motorola S37 文件。选择目标文件类型之前，应在 Option 属性页内同时选择 Makefile、Ary file、Generate listing file；

Include Interrupt Vector Table： 选中该项，在链接过程里包含中断向量表；

Include Start-Up Code ：选中该项，在链接过程里包含缺省的启动程序；

Align all resource with ：根据输入的数据把所有资源对齐；

Generate Initial Table：选中该项，在链接过程里产生一个初始化表；

External Symbol Files：需要链接在工程里的另外的符号表文件(*sym)；

Library modules： 指定和显示当前工程内的所包含的库模块。

[Section]：

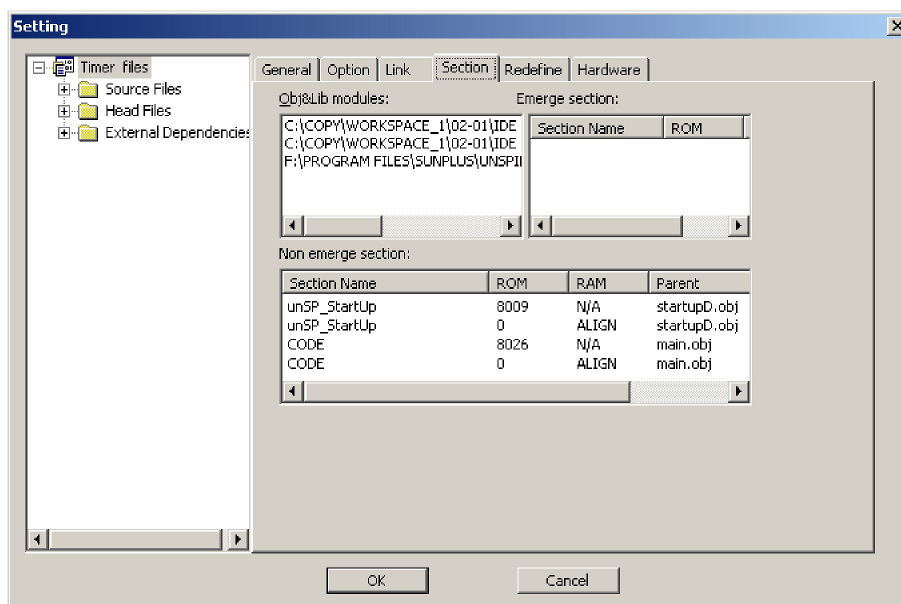


图 31　Section 属性页

在 Section 页里，显示当前工程中所有目标模块、库模块、合并段与非合并段，且可以设置当前工程的非合并段的地址、定位基址。

Obj & Lib modules：显示当前工程里的所有 obj 和 lib 文件；

Emerge section：　显示当前工程里的所有合并段；

Non-emerged section：显示当前工程里的所有非合并段。可以双击列表框内的 ROM 栏来改写这些段的地址或定位基址。在重链接工程后，这些指定段均会被定位到由定位基址引导的合适的地址上。

[Redefine]：

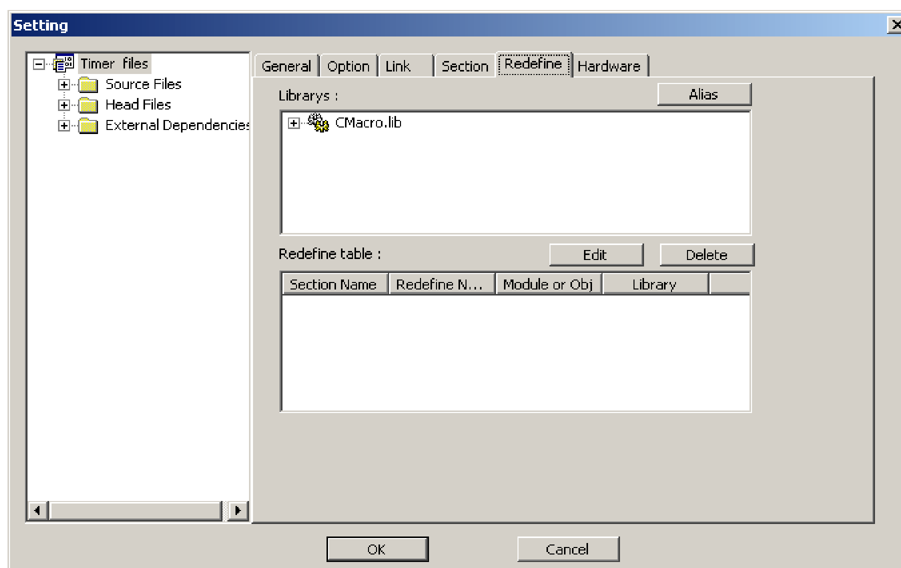


图 32　Redefine 属性页

在 Redefine 页里，可对库文件进行重定义。

Alias： 在 Librarys 列表框内选择某个段，再选择 Alias，改变当前段的名称；

Edit：被改变名称的各段的数据被列在 Redefine table 列表框内。用户选择某一行，再选择 Edit，也可选择双击这一行，再次改变段的名称。

Delete：删除 Redefine table 列表框内的内容。

[Hardware]：

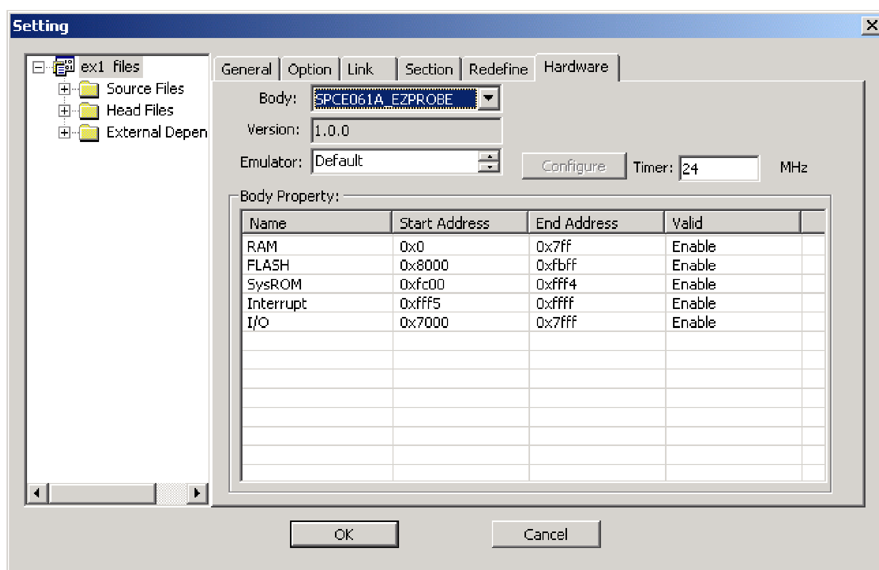


图 33 Hardware 属性页

在 Hardware 页里，可以设置µ'nSP®系列芯片的一些硬件信息。

Body：选择 Probe 型号。链接器和仿真器根据芯片的设置来进行链接和仿真；

Version：标识当前使用的 Simulator 和 ICE 的动态连接库的版本信息；

Emulator：根据 Probe 型号选择周边设备的仿真程序。这些程序实际为指定在 cpt 文件里的动态链接库；

Timer：设定系统时钟；

Configure：针对 Probe 型号的周边设备的仿真工具的设置；

Body property：显示内存映射结构。

设置文件选项

可以对编译器指定参数。选择 [Project]→[Setting]，打开 Setting 对话框，从左侧窗口内选择一文件，用户可设置 Compile 属性：

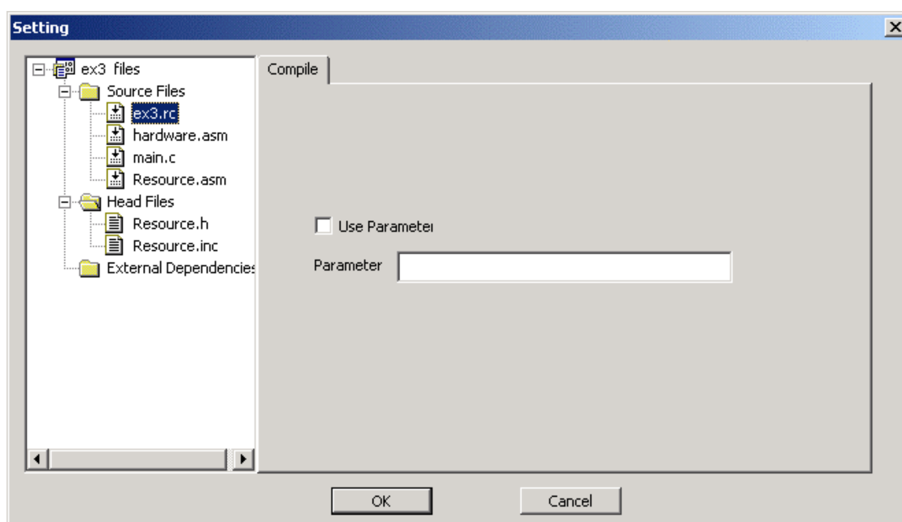


图 34 Setting 对话框

选择 Use Parameter，然后在 Parameter 框内输入参数。

编制工程

编制工程是对工程中的程序进行编译，然后把编译生成的二进制代码和库文件模块链接在一起，生成可执行目标代码文件和符号表文件的过程。

利用µ'nSP® IDE，可以对工程中的程序和库进行编制（Build）和重新编制（Rebuild）。选择[Build]→[Build]，µ'nSP® IDE 只处理本工程中截止到上一次工程编制完成后又经编辑过的文件。选择[Build]→[Rebuild]，µ'nSP® IDE 要重新处理工程中的所有文件。当用户的程序改动之处不多时用 Build 操作会为其节省大量的程序编译时间，使编制工程的效率大大提高。

创建工程后，µ'nSP® IDE 为工程设置缺省的选项。

编制工程之前首先要确定工程类型。Debug 版工程含有足够的符号调试信息，这些调试信息仅供µ'nSP® IDE 的调试器使用。Release 版工程则不含有任何符号调试信息。

每个工程都有指定的临时文件和目标文件的存储路径。

编制工程

过程：

选择 [Build]→[Compile]，对当前文件进行编译；
选择 [Build]→[Build]，编制工程；
选择 [Build]→[Rebuild All]，重新编制当前工程目标，将处理当前工程中的所有文件；
选择 [Build]→[Stop Build]，停止当前工程目标编制。

编制过程里产生的各项信息均显示在 Output 窗口的 Build 页内。包括程序编制过程里产生的一些错误和警告信息。没有显示错误，说明通过了编译链接，用户可进行程序的调试。在等待文件编制的过程中，可利用μ'nSP® IDE 完成其它操作。不过，此时有些菜单命令和工具栏中的按钮无效。

当用户选择 [Build]→[Stop Build]后，μ'nSP® IDE 首先要试图阻止当前工具的运行。若阻止不了，则会等待当前工具运行结束才终止编制过程。

运行程序

完成工程的编制后，用户可运行所编写的程序了。

加载程序

选择[File]→[Load Program]，将用户程序的可执行代码载入到内存内。

封装工程

利用μ'nSP® IDE 的封装功能，用户可以把当前工程的所有数据封装成一个 pak 文件。

创建封装文件
选择 [File]→[Pack Project]，打开 Save As 对话框：

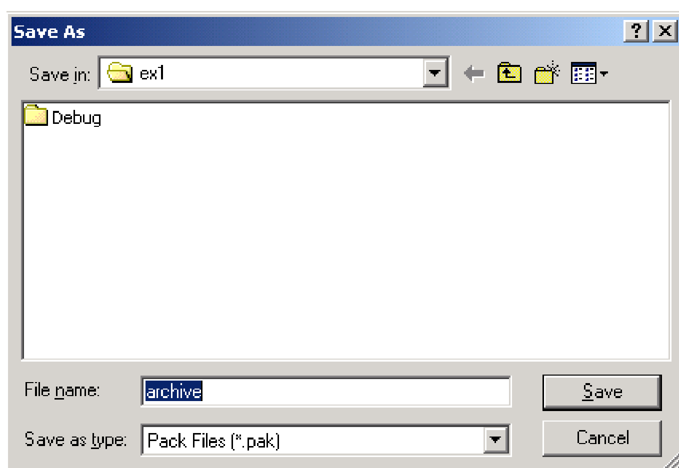


图 35 Save As 对话框

在 File name：框内输入封装文件名称，然后选择[Save]。 保存封装文件后，系统弹出一个 Select Pack File 对话框：

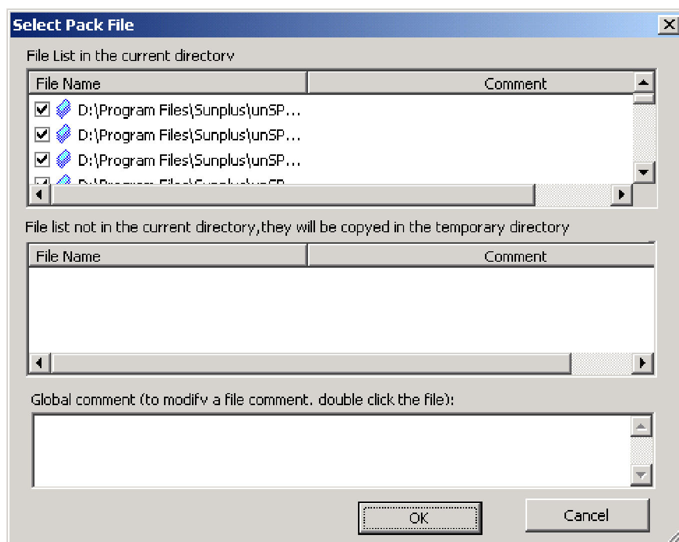


图 36 Select Pack File 对话框

该窗口前两个列表内，可选择需要封装的文件。

所有非本工程路径内的文件在进行工程封装后将被拷贝到临时目录内。

在 Global comment 框内，用户可指定所有文件的注解。当然，用户可为单个文件写注解，方法为双击前两个列表框内的文件名称，然后，在文本框内写注解。选择 [OK]，生成 pak 文件。

解压缩封装文件

选择 [File]→[Open Project]，打开一 pak 文件。此时，显示被封装的所有文件及相关注解。



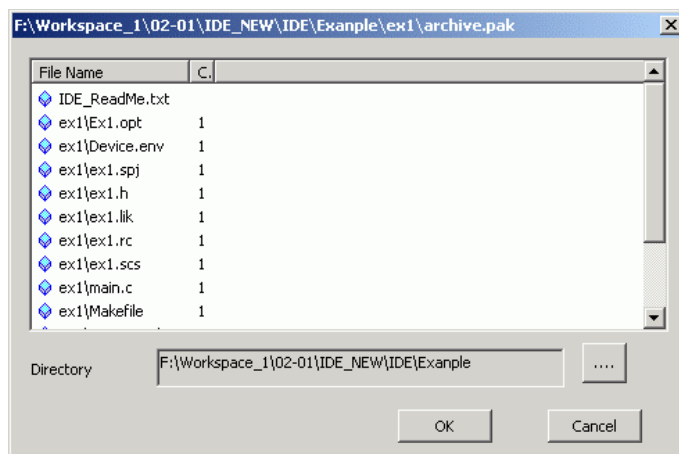


图 37 Pak 对话框

在 Directory 文本框内指定解压缩路径，选择[OK]。

第5章　代码控制

创建一个文件

利用µ'nSP® IDE，用户可创建 Inc(包含文件)、Asm(汇编文件)、C、CPP、CXX、H、HPP、HXX、 Text 等文件。对于当前版 IDE，在编译和链接的过程中不支持 CPP、CXX、HPP、HXX 文件。

选择 [File]→[New]，打开 New 对话框：

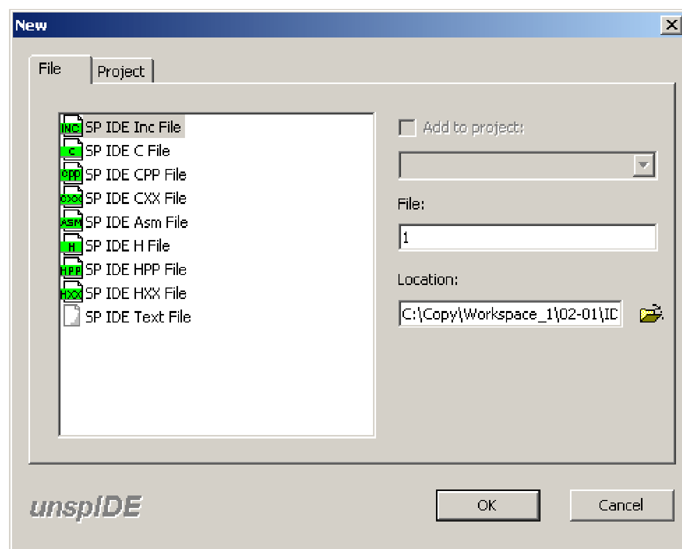


图 38　New 对话框

2. 选择 File 标签；
3. 在左侧框内选择文件类型。选择 Add to project，可把当前文件添加到某个工程内；
4. 在 File 文本框内输入文件名称；
5. 在 Location 文本框内输入文件的路径；
6. 选择 [OK]。

打开一个文件

选择 [File]→[Open]，打开 Open 对话框，在对话框内，用户可选择以文本和二进制格式打开文件。

第二种方法：
选择 [File]→[Recent Files]，可打开最近存取过的文件。在这个菜单内，最多可看到 8 个文件。

保存一个文件

当正在被编辑的文件的标题栏上文件名称后有“*”时，表示本文件正在被编辑。这时可用三种方法保存文件：
选择 [File]→[Save]来保存文件；
选择 [File]→[Save As]，给当前文件取一个新的文件名称；
选择 [File]→[Save All]，所有当前打开的文件都可被保存。

文本编辑器

书签

在文本编辑器内，可以在文件的任意行插入一个书签作为标志，以便用户查找。

设置书签
(1) 把光标移到指定的行，选择 [Edit]→[Bookmark]，在当前行设置一个书签，书签(蓝色的圆圈)显示编辑窗口的左侧。µ’nSP® IDE 提供了用热键设置书签的方法，即使用 Ctrl+n(n=0~9)；
(2) 选择 [Edit]→[Previous Bookmark]，查找到前一个书签所在行；
(3) 选择 [Edit]→[Next Bookmark]，查找到后一个书签所在行。

删除书签
(1) 把光标移到书签所在行处，选择[Edit]→[Bookmark]删除。
(2) 选择 [Edit]→[Clear All Bookmark]，清除所有的书签。
(3) 关闭当前文件，也可以清除所有的书签。

查找文本

在文本编辑器内，可以利用三种方法查找文本。

在当前文件内查找
(1) 选择 [Edit]→[Find]，打开 Find 对话框；
(2) 在 Find What 框内输入要找的字符串；
(3) 选择 Match Case 和 Match Whole Word Only 选项，用于设置大小写匹配和全字匹配；
(4) 选择 Up 和 Down 选项设置查找的方向；
(5) 选择 [Mark All]，用书签标识所有匹配的字符串所在行，选择 [Unmark All]，消除匹配的字符串所在行的书签；
(6) 选择 [Find Next]，顺着查找方向找到另一个匹配的字符串；
(7) 选择 [Edit]→[Find Next]和[Find Previous]，可向前和向后查找另一个匹配的字符串。

替换

(1) 选择 [Edit]→[Replace]，打开 Replace 对话框；
(2) 在 Find What 框内输入需要被替换的字符串 1，在 Replace With 框内输入新的字符串 2，字符串 2 可以是空的，这时按[Replace] 字符串 2 将替换字符串 1；
(3) 选择 Match whole word only 和 Match case 选项，用于设置大小写匹配和全字匹配；
(4) 选择 Selection 和 Whole file 选项，设置操作的范围；
(5) 选择[Find Next]，忽略当前匹配的内容。选择 [Replace]和[Replace All]，用来对当前和所有匹配的文本进行操作。

在多个文件内查找文本
(1) 选择 [Edit]→[Find In Files]，打开 Find in Files 对话框：

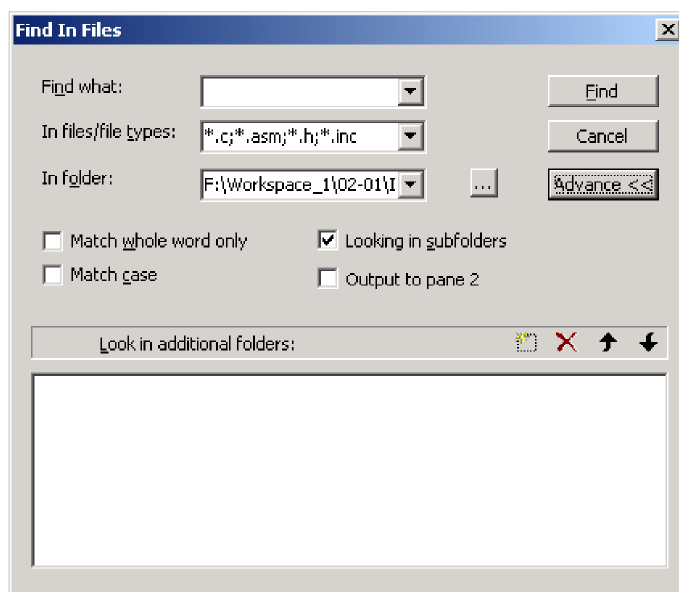


图 39 Find in Files 对话框

(2) 在 Find What 框内输入要找的字符串；
(3) 在 In files/file types 内选择文件类型；
(4) 在 In folder 内选择文件夹，也可利用浏览按钮选择路径；
(5) 选择 Match whole word only 和 Match case 选项，用于设置大小写匹配和全字匹配；选择 Looking in subfolders，在指定文件夹及其内所有子文件夹中查找；选择 Output to pane 2，指定在 Output 窗口的 Find in Files2 页内显示查找的结果；
(6) 选择 [Advance>>]，可以设置更多的查找路径；
(7) 选择 [Find]。

二进制编辑器

打开二进制文件

选择 [File]→[Open]，打开 Open 对话框；
在 Open as 内选择 Binary；

在文件列表内选择文件；
选择 [OK]。

注意，当其它编辑器正在对当前文件进行编辑时，请先关闭正在工作的编辑器再使用二进制编辑器。

编辑二进制代码

在已打开的二进制文件内选择某段代码；
直接更改代码。

第6章 资源

资源标识符

资源标识符由一串字符组成。向工程添加一个资源后，系统为资源自动分配一个标识符 RES_* (*：资源文件名称)。

编辑资源

添加资源

在 Workspace 窗口的 ResourceView 视窗内，选择一个元组，击右键，可以激活热键菜单，在热键菜单内选择[Add Files to Folder]，即可添加一个资源到当前元组。

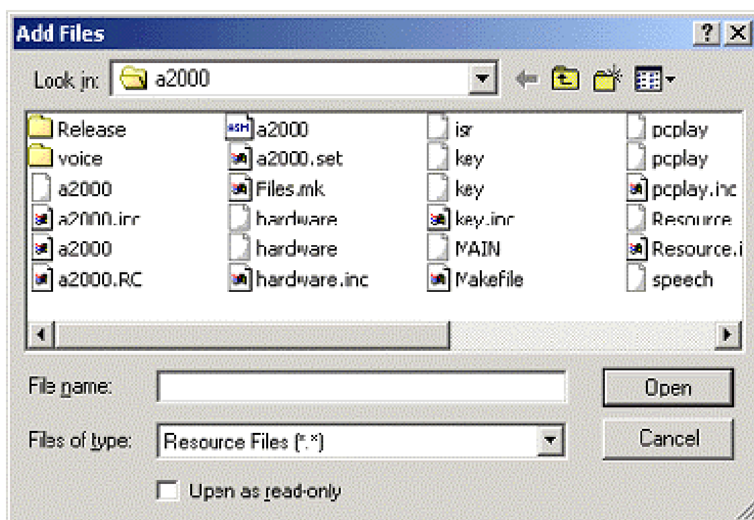


图 40 Add Files 对话框

改动工程内的资源

在 Workspace 窗口的 ResourceView 视窗内，选择某资源，击右键，可以激活一个热键菜单。

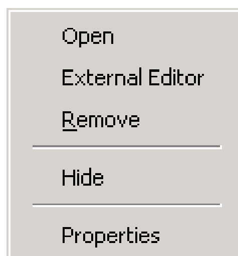


图 41 ResourceView 热键



[Open]：
用二进制编辑器打开所选择的文件。

[External Editor]：
用外部编辑器打开所选择的文件。

[Remove]：
删除所选择的文件。

[Hide]：
隐藏 Workspace 窗口。

[Properties]：
在 Resource Properties 对话框内，可以改变资源的存储位置(Resource file)、资源标识符(Resource ID)、输出符号表(Export Symbols)，选择相应的过滤器(Filter)。过滤器设置好后，可在链接之前将资源转换为 Sunplus 格式。

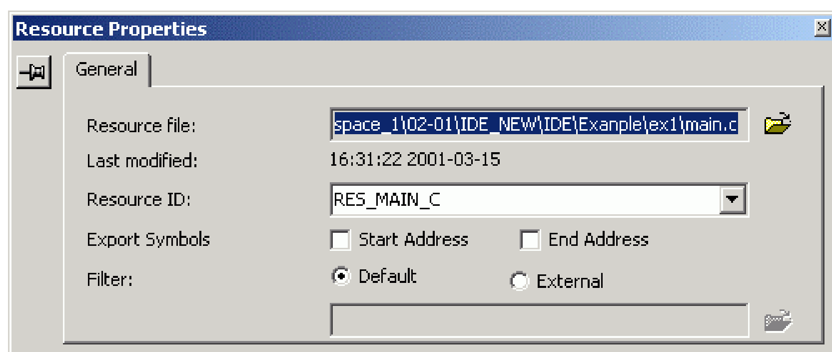


图 42 Resource Properties 对话框

使用外部过滤器的格式：
Filter.exe <输入文件> <输出文件 sp>　　输出文件的扩展名必须为 sp

第7章　调试器

利用µ'nSP® IDE，用户可以轻松地找到程序的逻辑语法错误，借助一些调试窗口来查看变量、寄存器和内存等的状态来监视程序的运行，同时可以连续、单步和设置断点等运行方式来运行程序。

控制程序运行

控制程序运行的目的是为了迅速查找程序中存在的错误。用户可根据程序调试的状态、结果和错误目标搜索的需要来选择合适的控制方法。

首先，选择 [Build]→[Start Debug]→[Download]，把用户程序的可执行代码载入到内存。

在调试状态下，可以利用 Build 菜单里的 Start Debug 子菜单内的各项进行调试。Start Debug 子菜单内各命令和对程序运行的控制操作如下：

Go
从当前程序指针处运行程序，直到遇到断点/程序结尾处停止运行。

Restart
从起始地址重新运行程序。

Stop Debug
终止程序的调试，返回到编辑状态。系统会自动保存所有调试状态的设置，以便下次调试再用。

Break
中止程序运行。

Step Into
单步运行程序，当下条指令是函数时会进入被调用函数继续单步执行。

Step Over
单步运行程序，当下条指令是函数时会将被调用函数整体当作一步执行完，然后接着单步执行函数指令的下一条指令。

Step Out
单步运行程序，当下条指令是函数时会从被调用函数执行程序指令直至返回主程序的指令处继续单步执行。

Run to Cursor
从当前指令行执行到光标所在指令行为止。

调试窗口

调试状态下，View 菜单里的各调试窗口可以被激活。

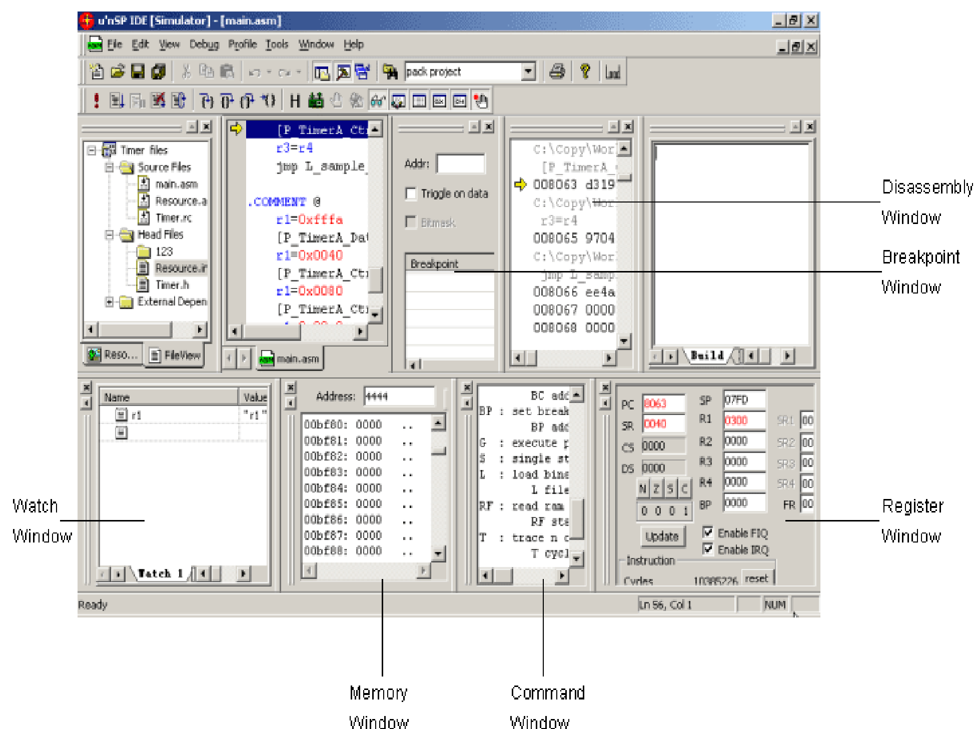


图 43 调试窗口

内存窗口(Memory Window)：
显示或修改内存单元的内容。该窗口上方的 Address 文本输入框可直接指定一个内存单元地址，实现快速查看功能。

寄存器窗口(Register Window)：
显示 CPU 各寄存器,中断标志,指令周期数的值。

命令窗口(Command Window)：
在该窗口内直接输入命令控制程序的状态。

断点窗口(BreakPoint Window)：
显示或设置断点信息。

变量窗口(Watch Window)：
显示变量或表达式的值，用户可以输入变量名及编辑表达式。

反汇编窗口(Disassembly Window)：
显示内存中程序的反汇编代码。通过双击鼠标左键或 Ctrl + G 热键可激活 Goto Address 对话框，实现快速跳转到指定地址的反汇编行的功能。

缓存区窗口(History Buffer Window)： 显示已执行的指令、状态等信息。

第8章 剖析器

剖析器功能

剖析器是μ'nSP® IDE 提供的一个功能强大的分析工具。一旦通过编制确定了程序代码，就可在调试程序过程中应用此工具来剖析、优化程序代码。具体地，此工具具有以下一些功能：

提供代码优化的准确信息。包括某段程序花费了多少个指令周期的执行时间、程序的标号流、中断请求等一些有助于提高程序效率的信息；
检测和分析程序运行过程中使用的算法的有效性；
检查用户程序的代码段是否被测试程序覆盖。

在运行剖析器之前，请停止程序的调试和运行。

剖析器操作

选择 [Build]→[Profile]，打开 Profile Configure 对话框：

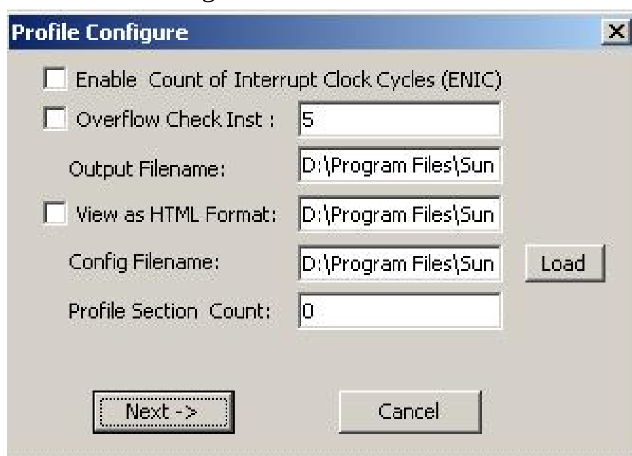
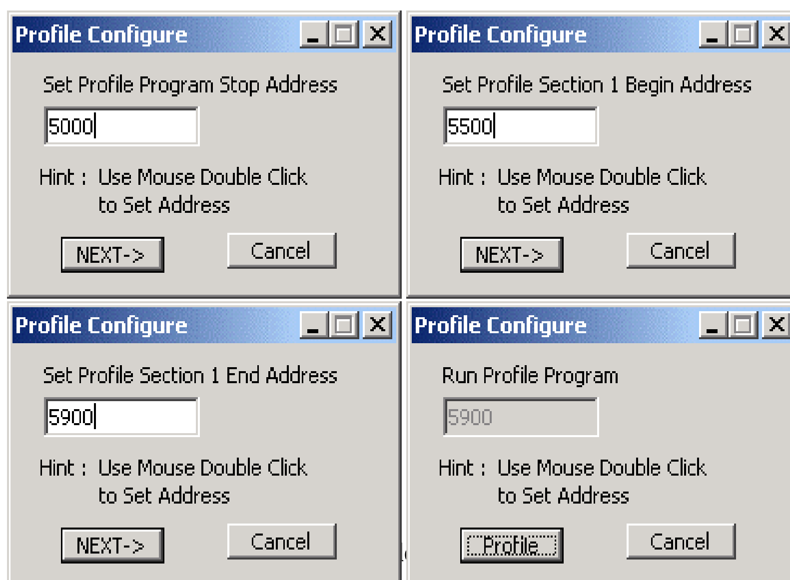


图 44 Profile Configure 对话框 1

(1) Enable Count of Interrupt Clock Cycles (ENIC)：
选择 ENIC，以便在 IRQ 中断服务子程序中仍可连续剖析代码。若要求剖析的代码段处于 IRQ 子程序中，请选择此项。
(2) Overflow Check Inst：
设定当运算产生溢出时多少个指令内未检查溢出标志便产生警告信息。
(3) Output Filename：
指定容纳产生剖析结果的文件的名称。
(4) Config Filename：
指定存储配置参数的文件名称，用于剖析程序时重新装入。
(5) Profile section count：

指定需剖析的程序段的段数，N 值不超过 32

2. 选择 [Next]按钮，出现如下图的对话框。
在对话框的文本框内，用户可输入地址， 也可以直接在源程序文件中双击鼠标左键来指定程序剖析开始和结束的地址。



3. 选择 [Profile]按钮来启动剖析器。剖析器执行速度受到剖析区域的大小和存储剖析结果的文件路径等因素的影响。用户可在屏幕上打开一个剖析文件(*out)查看剖析结果。

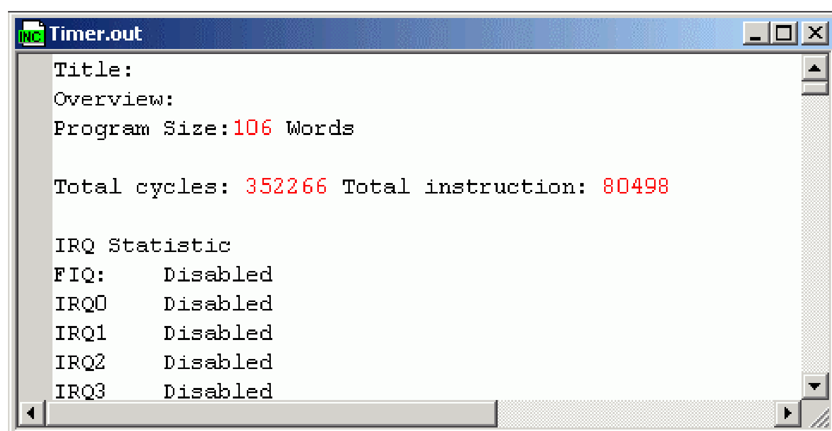


图 46 History Buffer 窗口

第9章 工具

定制开发环境

当用户对开发环境越来越熟悉时，越希望开发环境能符合自己工作的风格。

开发环境的很多行为一旦设置便会被“记忆”下来，在下一次启动时它会成为缺省行为。如在调试状态下使用的某些显示窗口和工具条的状态均会被记录下来，用于下次进入调试时恢复使用。

Tools

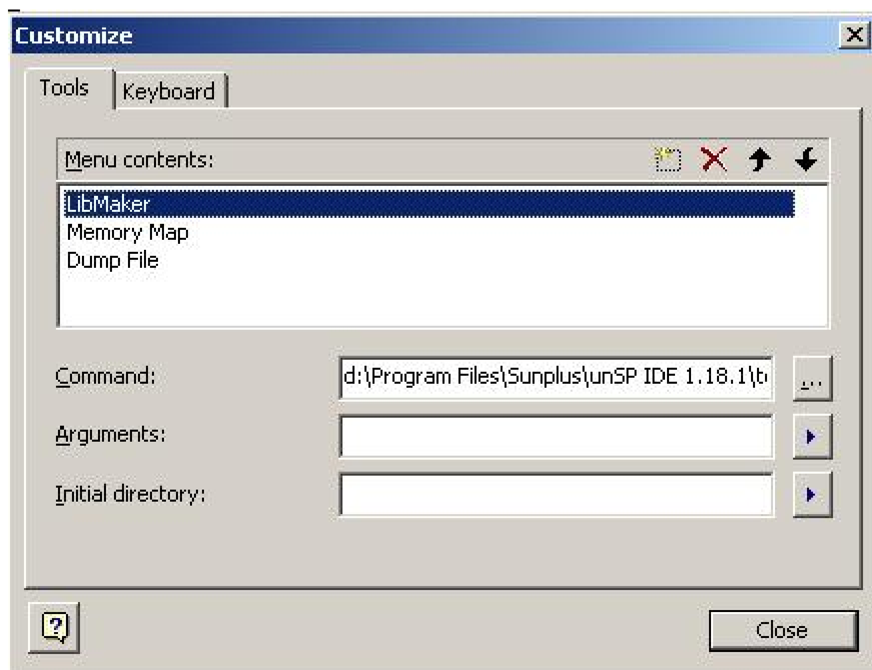


图 47　Tools 属性页

在该页，用户可设置需要使用的外部工具。设置的工具标识将显示在 **Tools** 菜单内。此时，用户就可以在集成开发环境内使用该工具。

设置外部工具的时候，可根据需要添加参数。参数说明在以后的内容里加以描述。

向 Tools 菜单添加工具

选择 **Menu Contents** 框上部的添加按钮 ，或双击 **Menu Contents** 框内部的空行，来添加新工具。输入工具名称，按[ENTER]键；
选择刚输入的工具名称；
在 **Command** 框内输入可执行程序的路径和程序名称，也可通过浏览按钮来设置。例如：**C：\WINDOWS\NOTEPAD.EXE**；

在 **Arguments** 框内输入传递给外部工具所使用的参数；
在 **Initial** 框内输入可执行程序的默认路径。

通过对 **Menu Text** 进行编辑，用户可改变工具的标识。

Arguments 和 Initial Directory 的各个参数如下：

参数名称	宏	功能
File Path	$(FilePath)	当前源文件的完整名称(驱动器＋路径＋文件名称)，如没有源文件窗口被打开，该项为空
File Directory	$(FileDir)	当前源文件的目录(驱动器＋路径)，如没有源文件窗口被打开，该项为空
File Name	$(文件名称)	当前源文件的文件名称(文件名称)，如没有源文件窗口被打开，该项为空
File Extension	$(FileExt	当前源文件的文件扩展名
Current Line	$(CurLine)	在活动窗口内当前行的位置
Current Column	$(CurCol)	在活动窗口内当前列的位置
Current Text	$(CurText)	当前光标所在文本的位置
Current Directory	$(CurDir)	当前工作路径(驱动器＋路径)
Target Path	$(TargetPath)	目标完整名称(驱动器＋路径＋文件名称)
Target Directory	$(TargetDir)	目标路径(驱动器＋路径)
Target Name	$(TargetName)	目标名称(文件名称)
Target Extension	$(TargetExt)	目标扩展名
Workspace Directory	$(WkspDir)	工作区目录(驱动器＋路径)，如没有活动工作区，该项为空

参数名称	宏	功能
Workspace Name	$(WkspName)	工作区名称(文件名称)，如没有活动工作区，该项为空
Project Directory	$(ProjectDir)	工程目录(驱动器＋路径)
Current Body	$(CurrentBody)	当前芯片类型

改变 Tool 菜单上的工具
在 **Menu Contents** 框内，选择要改变的工具，然后，用户可改变工具标识和相应的设定。

利用在 **Menu Contents** 框上部的各个按钮，用户可改变工具在 Tool 菜单上的位置，也可以删除工具。

Keyboard

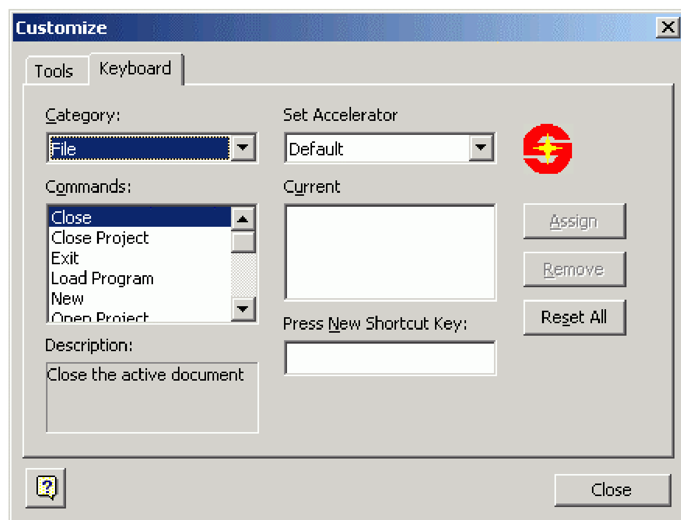


图 48 Keyboard 属性页

在该窗口，用户可为各个菜单命令设置、改变相应的热键，还可恢复各菜单命令的热键。

设置、改变相应的热键
在 Category 列表框内，选择菜单，以便在 Commands 框内显示菜单内的所有子菜单项；
在 Commands 列表框内，选择子菜单；
在 Press New Shortcut Key 框内，输入某个键/键的组合，然后选择[Assign]。注意，避免使用 ESC、CTRL+ALT+DEL 等操作系统占用的键；
在 Current 框内，选择已设定的热键，按[Remove]，删除该热键。

恢复各菜单的热键
选择[Reset]，恢复初始热键。

自定义编辑器

Options 对话框包含了对编辑器的形状、功能，包含文件和库文件的目录等的设置。

在 Options 对话框内，各项选项的设定对编辑窗口和调试窗口都有很大的影响。IDE 有四种布局：编辑模式下的标准布局、编辑模式下的全屏布局、调试模式下的标准布局和调试模式下的全屏布局。

Format

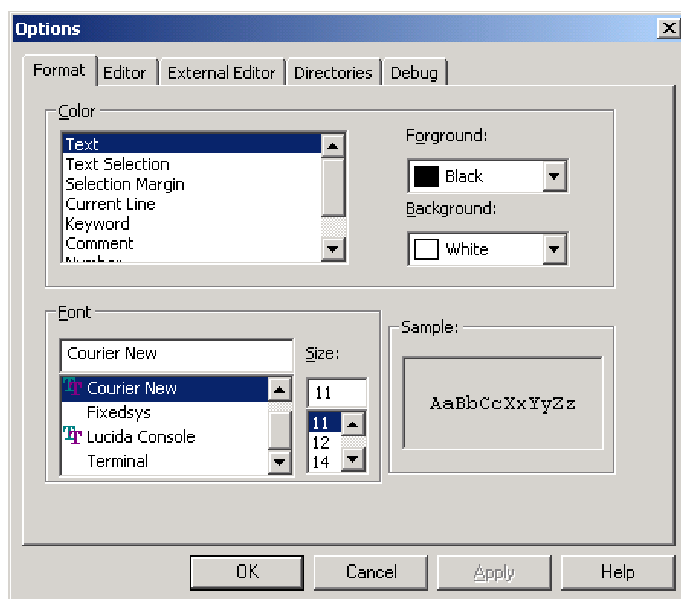


图 49 Format 属性页

在该窗口，可以设置文本编辑器和反汇编窗口的字体、字号、前背景颜色。

Editor

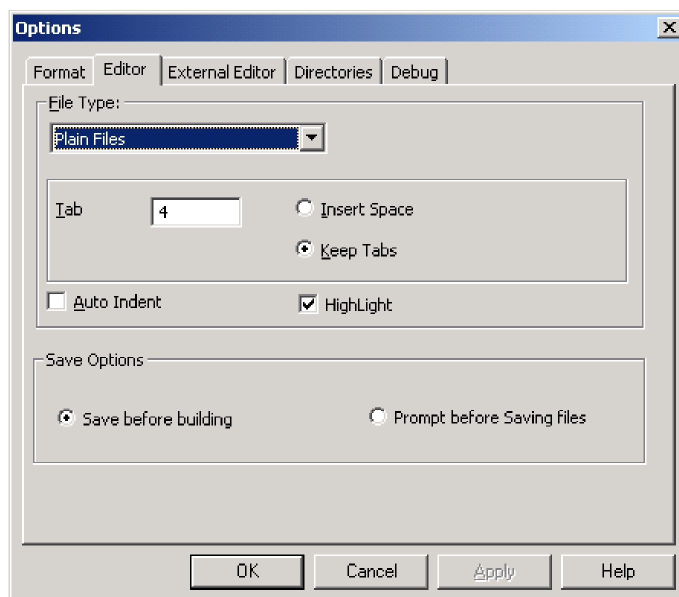


图 50 Editor 属性页

在该窗口，可以设置针对某类文件每按一次 Tab 键后光标移动的字符数、方式，缩进位置等。选择 Save before building/Prompt before saving file 来控制编制工程前文件采用的保存方法。

External Editor

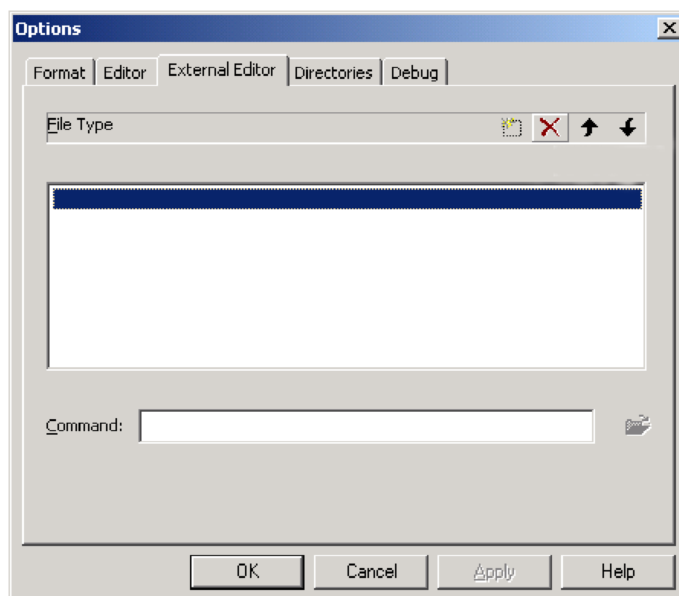


图 51 External Editor 属性页

在该窗口，用户可设置某类文件所使用的外部编辑器。设置后，可以在 Edit 菜单内使用外部编辑器。

Directories

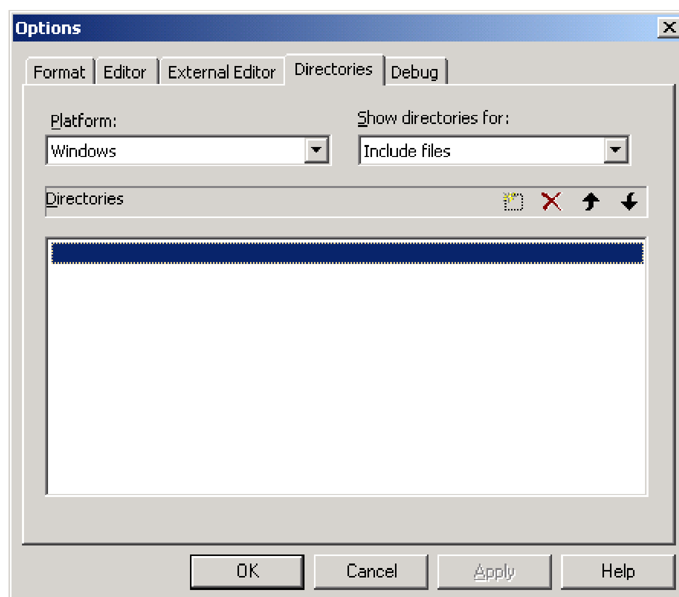


图 52 Directories 属性页

在该窗口，用户可设置整个环境所包含的目录。

Show directories 列表显示库文件和包含文件两种目录类型，Directories 列表显示用户添加的路径。

Debug

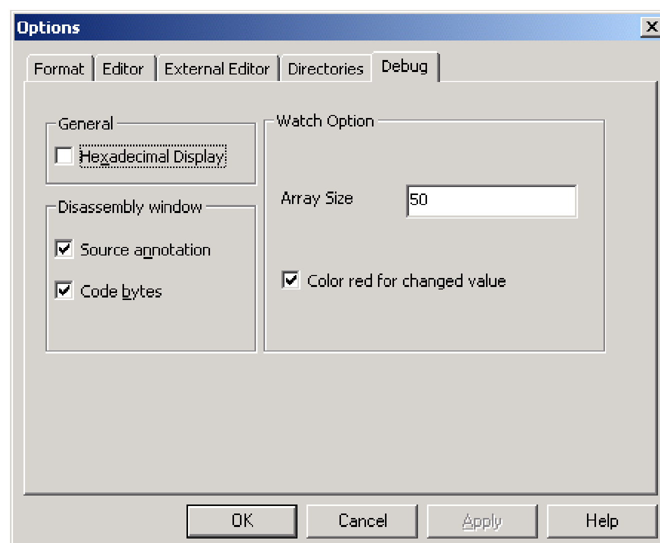


图 53 Debug 属性页

在该窗口，可以设置 Debug 窗口的各显示功能。

Hexadecimal Display：以十六进制格式显示内存的内容；

Source annotation：在 Disassembly 窗口内显示源代码；
Code bytes：在 Disassembly 窗口内显示指令的操作码；
Array Size：指定在 Watch 窗口内显示的数组的长度；
Color red for changed value：指定 Watch 窗口中被改动的数据用红色显示，选中选项将影响调试的速度。

其他工具

Dump File

利用 Dump File 工具，可以把指定地址范围内的数据转存到文件内。激活 Dump File 工具的方法：选择 [Tools]→[Dump File]。

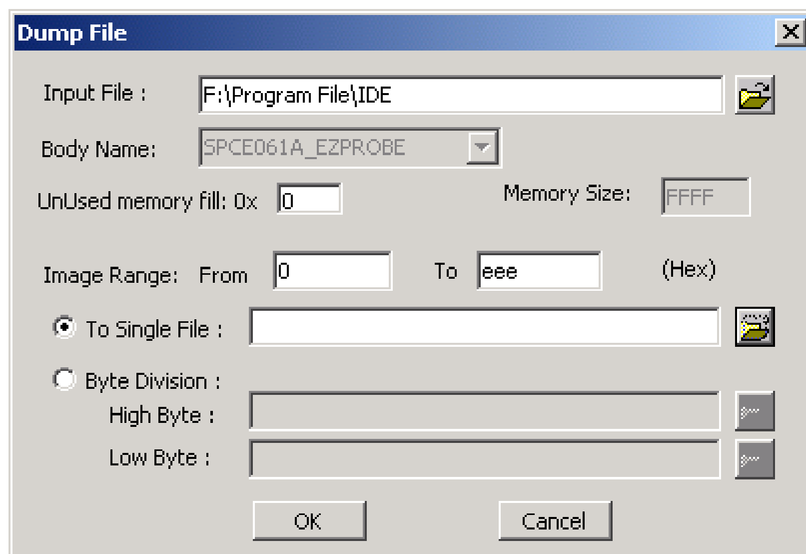


图 54 Dump File 对话框

Input File: 打开源文件，源文件支持 S37 和 TSK 两种；
UnUsed memory fill: 指定空白地址单元所填入的数据；
Image Range: 指定存储地址单元范围；
To Single File: 把指定存储地址单元范围内的数据转存到一个文件内；
Byte Division: 按高地址和低地址把指定存储地址单元范围内的数据转存到两个文件内。

LibMaker

利用 LibMaker 工具可创建一个库或对库进行改动。激活 LibMaker 工具的方法：选择 [Tools]→[LibMaker]。

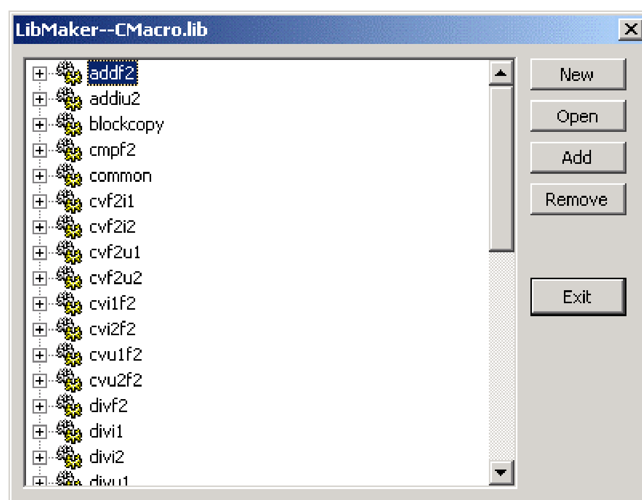


图 55 LibMaker 对话框

选择 [New]，创建一个库；

选择 [Open]，打开一个库；

选择 [Add]，向库内添加目标文件；

选择目标文件，用[Remove]可以删除；

选择 [Exit]，退出窗口。

MemoryMap

利用 MemoryMap 工具，可以查看内存的利用情况、标号等。激活 MemoryMap 工具的方法：选择[Tools]→[MemoryMap]。

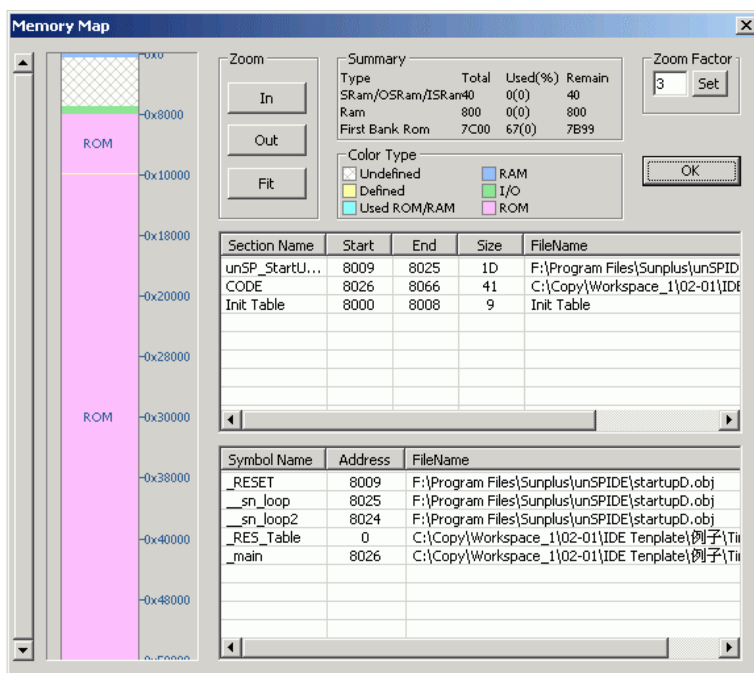


图 56 MemoryMap 对话框

选择 [In]，放大映象；

选择 [Out]，缩小映象；

选择 [Fit]，调整映象使映象尺寸和窗口大小相适合；
[Zoom Factory]，设置映象缩放因数；
选择 [OK]。

第10章 外设仿真器

Default

选择[Project]→[Setting]→[Hardware]→[Default]，启动 IDE 周边环境的默认设置。

AD Converter

AD converter 是一个用于把接收到的模拟信号转换成数字信号的工具。AD converter 的输入数据有两种形式：周期性波形和非周期性波形。

选择[Project]→[Setting]→[Hardware]→[Emulator]→[AD]→[Configure]，启动 AD Converter。

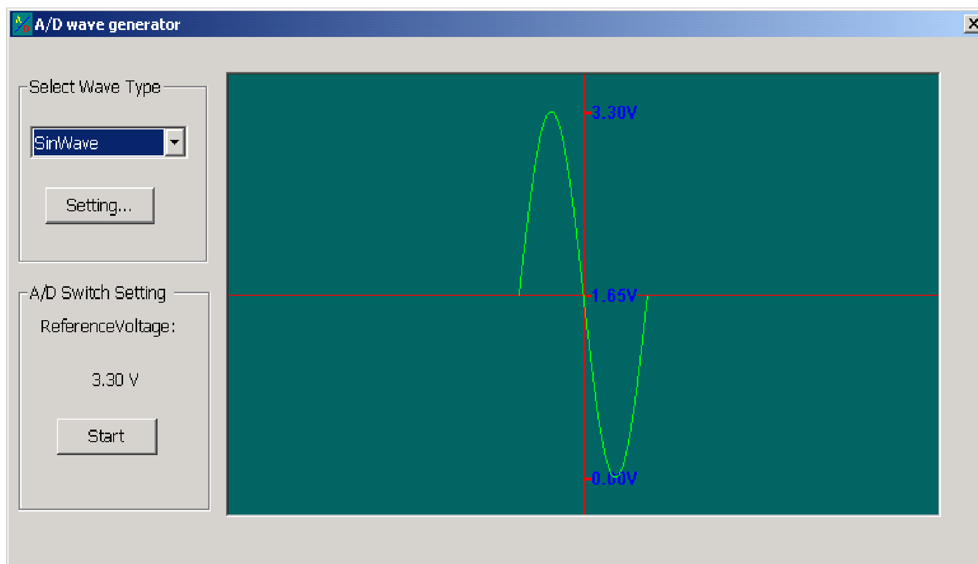


图 57 AD Converter 对话框

设置各选项，然后选择[Start]键，把模拟信号转换成数字信号。

周期性波形

正弦波形
在 Select Wave Type 内选择 SinWave， 打开 Setting Dialog 对话框：

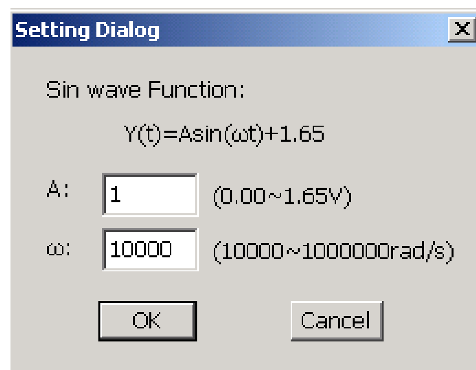


图 58 SinWave

A：振幅

ω：角速度

方波

在 Select Wave Type 内选择 RectangleWave， 打开 Setting Dialog 对话框：

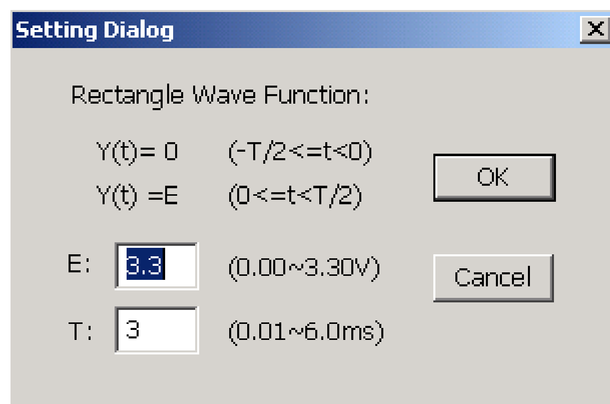


图 59 RectangleWave

E： 电压

t： 采样时间周期

T： 周期

锯齿波

在 Select Wave Type 内选择 SawtoothWave， 打开 Setting Dialog 对话框：

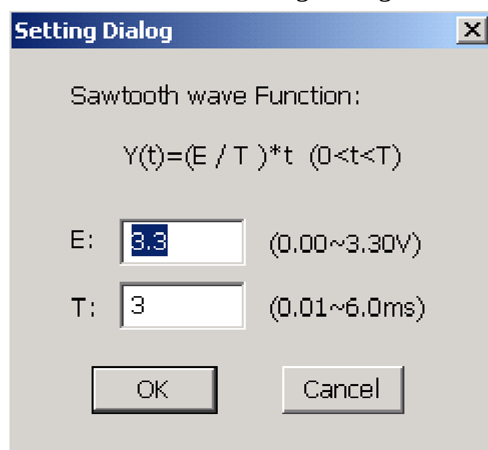


图 60 SawtoothWave

E： 电压

t： 时间

T： 周期

非周期性波形

在 Select Wave Type 内选择 CustomType， 打开 Setting Dialog 对话框：

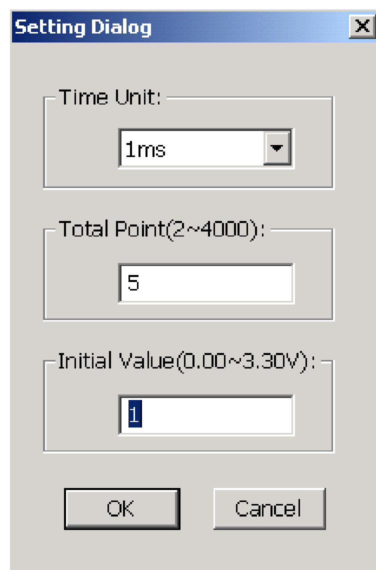


图 61 CustomType

Time Unit：每两点之间的时间间隔，包括 1ms、10ms、100ms 和 1000ms；

Total point：波形内总的点数(最大数：4000)；

Initial Value：初始电压(0V~+3.3V)。

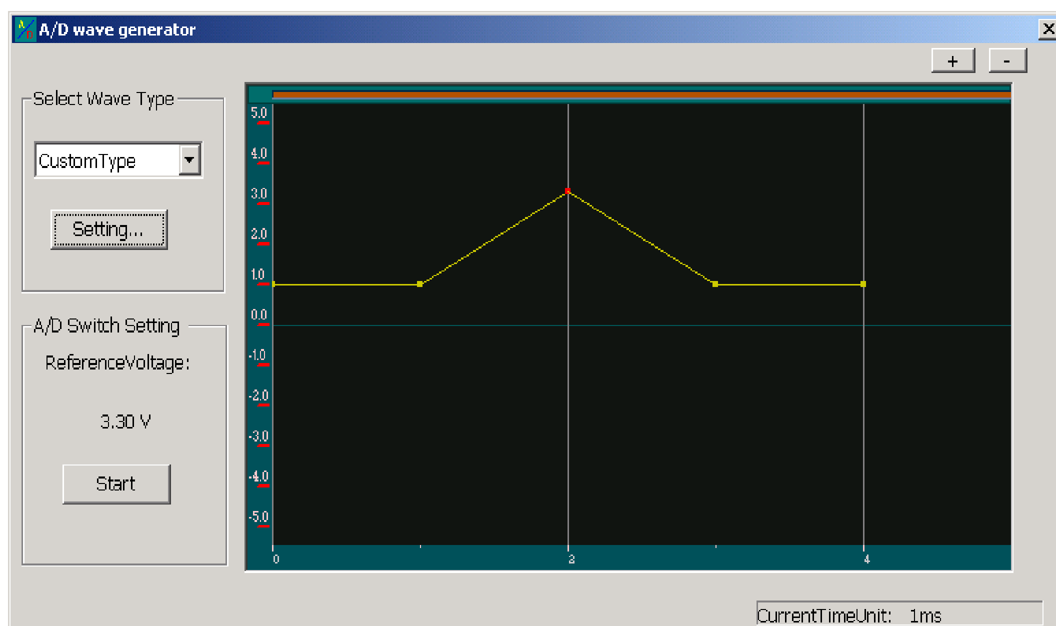


图 62 AD Wave Generator

数轴横坐标为时间，纵坐标为电压，电压的单位可以在 CurrentTimeUnit 栏位内看到。

设置电压有两种方法：
把光标移到一个点上，这时，所在点相应的时间单位和电压值显示在弹出窗口内。使用鼠标拖动这个点，即可以改变它的电压值。

拖动鼠标选择某区域，然后点鼠标右键，打开 Setting Dialog 对话框：

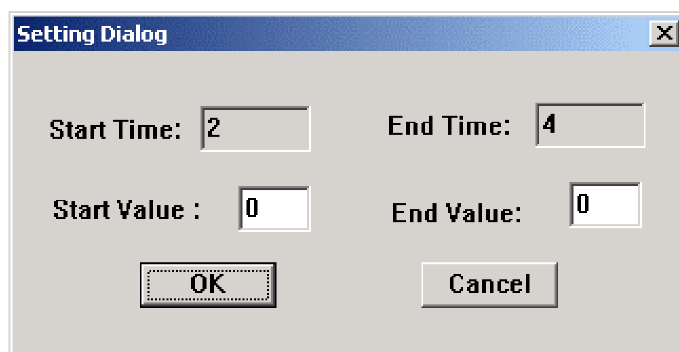


图 63 Setting Dialog 对话框

Start Time：区域的起始时间
End Time： 区域的结尾时间
Start Value：起始点电压
End Value： 结尾点电压

PWM(Pulse Width Modulation)

PWM Wave 窗口是数字示波器，把数字信号转换成模拟信号，声音数据通过 PWM 端口输出到 PWM Wave 窗口。

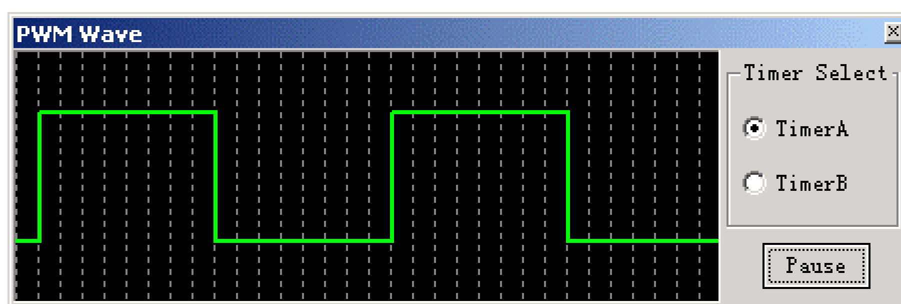


图 64 PWM Wave 窗口

Input_Output

Input_Output 工具用于仿真硬件输入输出系统的功能。

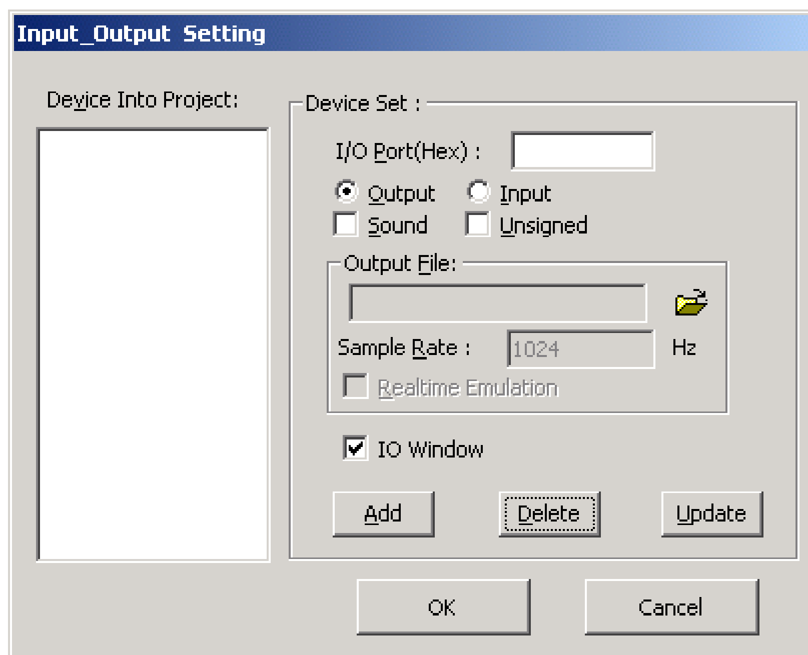


图 65 Input_Output 对话框

I/O Port [Hex]：设备端口地址。运行程序时通过该端口可以读写数据；

Output and Input： 输入输出方向；

Sound and Unsigned： 选择文件类型。选择 Sound，向文件加入一个 Wave 文件头，文件可作为*.wav 文件被播放。选择 Unsigned，所有输出的数据都被设置为无符号数据；

Sample Rate：声音采样率；

Real time Emulation：根据用户设置的采样率进行实时数据的存取；

IO Window：在运行过程里，通过 IO 端口存取的数据显示在 IO Window 内。

PortIO

选择[Project]→[Setting]→[Hardware]→[PortIO]，对 IO 口属性进行设置。



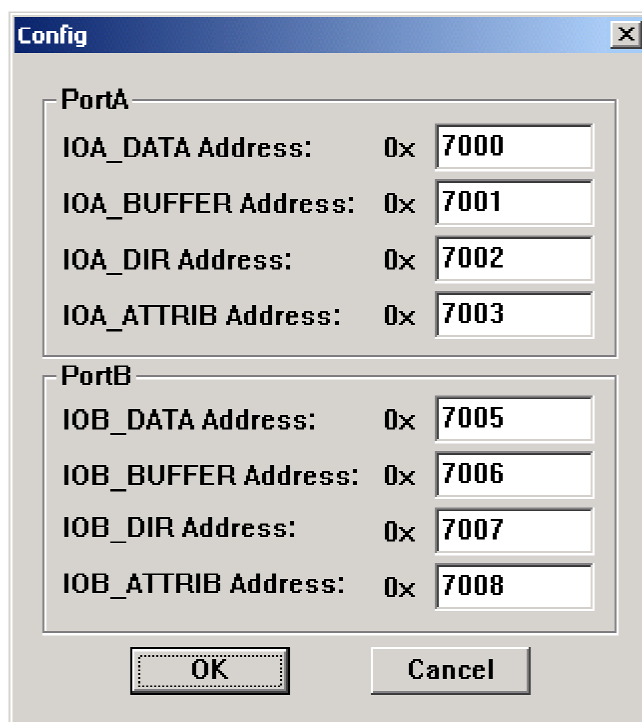


图 66

IOn_DATA Address：数据口位地址
IOn_BUFFER Address：数据向量口位地址
IOn_DIR Address：方向向量口位地址
IOn_ATTRIB Address：属性向量口位地址

第11章 热键

选择 [Help]→[Keyboard]，在 Keyboard 窗口用户可以看到所有的命令和热键。

第12章 答疑

1.为何有时不能正常运行?

(1)在生成工程的过程里，遇到像不能找到所需的可执行程序等情况时，请选择 [Project]→[Setting]，查看 CC、AS 和 LD 的设置是否准确、以下各文件是否存放在指定位置： PortIO.dll、Input_Output.dll、SIMDLL_ISA11.dll、make.exe、rm.exe、shell.exe、simulator_ISA11.exe、cpp.exe、cc1.exe、xasm16.exe、xlink16.exe、xlib16.exe 和 gcc.exe；

(2) 当µ'nSP® IDE 启动过程、状态切换过程、窗口显示不正常时，请运行 regedit.exe，删除 [HKEY_CURRENT_USER\Software\Sunplus\IDE180]注册信息，然后重启动µ'nSP® IDE；

(3)使用 Windows NT/2000 系统时，如果µ'nSP® IDE 在下载过程里出现异常，请查看系统目录/Win32 目录下是否有 giveio.sys，然后在安装目录下运行 giveio.exe。

2. 不能和 ICE 正常连接？

(1) 查看 ICE 的电源 LED 是否亮着，如果没亮，请查看电源正常接通与否；

(2) 查看 PC 是否通过打印机端口和 ICE 相连接；

(3) 查看打印机端口是不是处于标准模式。

3. 资源占用过多

如果µ'nSP® IDE 的图标和菜单不能正常显示，说明系统资源被占用得过多，请关闭某些正在运行的程序，然后重新启动µ'nSP® IDE。